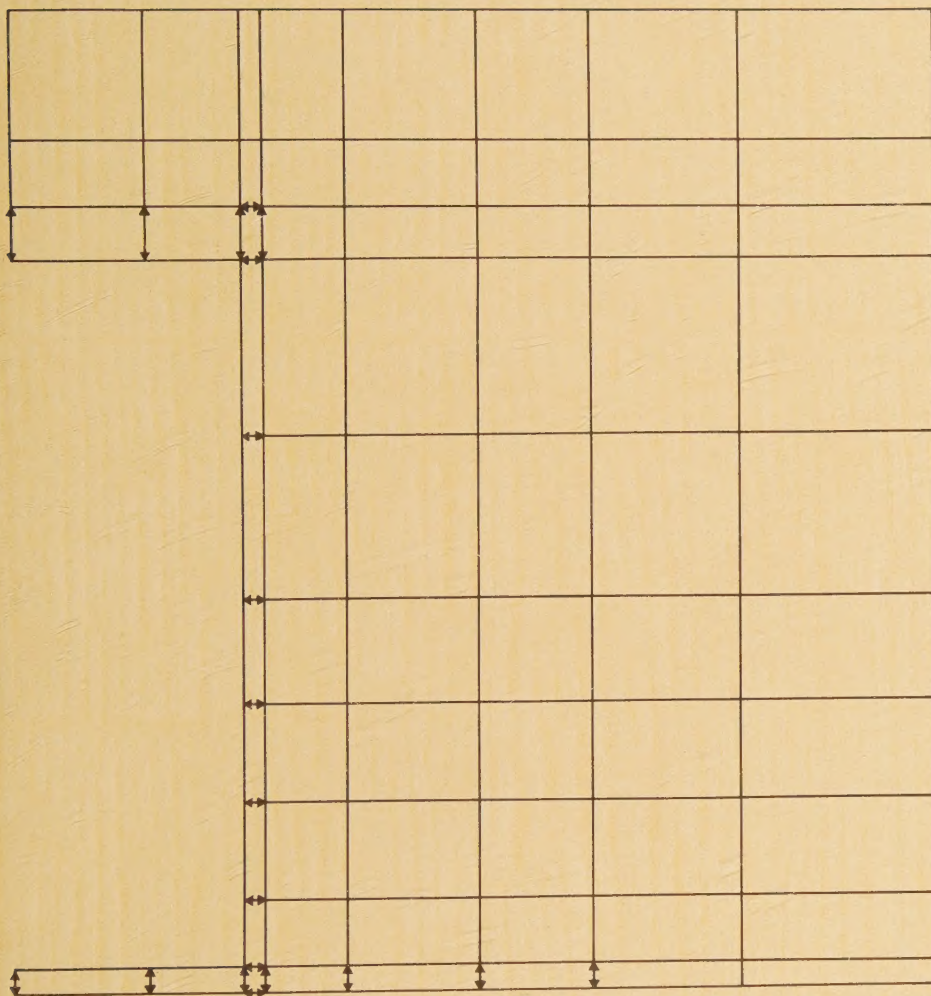




ULF WICKSTRÖM

TASEF-2 — A COMPUTER PROGRAM FOR
TEMPERATURE ANALYSIS OF STRUCTURES
EXPOSED TO FIRE

1979



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41553413_0027

LUND INSTITUTE OF TECHNOLOGY LUND SWEDEN

DEPARTMENT OF STRUCTURAL MECHANICS

REPORT No. 79-2

TASEF-2 - A Computer Program for Temperature Analysis
of Structures Exposed to Fire

Ulf Wickström

ACKNOWLEDGEMENTS

The people who work in the Division of Structural Mechanics and Concrete Construction here in Lund have helped to create an atmosphere in which I could finish the work reported in this dissertation. For this they deserve my heartfelt thanks.

Professor Ove Pettersson has for many years been both in the fore- and background of my work. As head of this Division and as advisor for my doctoral work, his example has helped me to define my goals as a research engineer.

Professor Hans Petersson has graciously and carefully reviewed the manuscript of this work again and again as it has evolved. Hans has the gift of making one believe that requests for advice are never an imposition and that he gains as much from any interchange as does the student. While I cannot believe this to be true in my case, I thank him both for his criticisms, but especially for his patience and time.

I also wish to acknowledge Judith Sanders for her proficient assistance in editing this report and her patience in encouraging and supporting me in my endeavor.

Tarja Aunola and Bo Zadig typed the report and drew the figures, and advised me on the design of the tables and figures.

The project has been partly sponsored by the National Swedish Council for Building Research.



TABLE OF CONTENTS

ACKNOWLEDGEMENT

TABLE OF CONTENTS	PAGE
1. INTRODUCTION	5
2. HEAT TRANSFER ANALYSIS	7
2.1 Basic Equations	7
2.2 Initial and Boundary Conditions	8
3. FINITE ELEMENT APPROXIMATION	12
3.1 Solution Techniques	12
3.2 Basic Approximations	12
3.3 Matrix Equilibrium Equations for Transient Heat Conduction	13
3.3.1 Conductivity Matrix	15
3.3.2 Heat Capacity and Volume Matrices	17
3.3.3 Internally Generated Heat	18
3.3.4 Boundary Heat Flow	18
3.4 Time Integration	21
3.4.1 Forward Differences	22
3.4.2 Critical Time Increment	24
4. COMPUTER PROGRAM	27
5. EXAMPLES	31
5.1 Example I - Square Plate Subjected to Heat Transfer from Surrounding Gas	31
5.2 Steel Beams Embedded in Concrete	32
5.2.1 Material Properties and Boundary Conditions	32
5.2.2 Example II - Wide-Flange I-beam	42
5.2.3 Example III - Box Girder	43
6. SUMMARY AND CONCLUSIONS	53
6.1 Present Study	53
6.2 Future Development	54

7.	REFERENCES	56
	APPENDIX A - USER'S MANUAL	59
	APPENDIX B - PROGRAM LISTING OF TASEF-2	76
	APPENDIX C - EXAMPLE INPUT	110

1. INTRODUCTION

A nonlinear heat flow equation must be solved to predict the distribution of temperature in a structure exposed to fire. Since analytical solutions of such equations exist only for idealized cases, numerical schemes that incorporate either the finite element or finite difference method have generally been employed to approximate heat conduction [1-5].

Ödeen computed temperature distribution in homogeneous concrete cross-sections exposed to fire [1] using a program based on the finite difference method. Latent heat due to evaporation of water was considered in the calculation, but only structures with simple geometries were analyzed. Based on work by Wilson et al. [2,3] the finite element programs FIRES-T [4] and later FIRES-T3 [5] were developed for analyzing thermal response of structures exposed to fire. An implicit backward difference time integration scheme is used in these programs. Computation therefore often becomes unnecessarily expensive, and materials with latent heat - for instance humid concrete - cannot be analyzed accurately.

In this report TASEF-2 (Temperature Analysis of Structures Exposed to Fire - Two Dimensional Version) a computer program based on the finite element method is described. Structures comprised of one or more materials and structures that enclose voids can be analyzed. Heat transferred by convection and radiation at the boundaries can be modeled. The explicit forward difference time integration scheme used in TASEF-2 facilitates consideration of latent heat in the calculation of temperature in materials such as humid concrete. The maximum length of the time increment that can be used without inducing numerical instability is discussed, and some procedures to avoid very short time steps are suggested. In the present version of the program two-dimensional rectangular elements are used; input of the geometry and generation of the finite element mesh have been automated.

In the report, the theoretical model and solution techniques are derived, the organization of the computer program is explained, and a commentary on practical aspects of using the program is made. Several examples are analyzed using TASEF-2 and calculated temperatures are in some cases compared to experimental results. The report contains fully annotated input instructions, and a listing of the program.

2. HEAT TRANSFER ANALYSIS

2.1 Basic Equations

The governing equations for heat conduction are the heat balance equilibrium equation

$$\underline{\nabla} \underline{q} + \dot{e} - Q = 0 \quad (2.1)$$

and the Fourier law

$$\underline{q} = -\underline{k} \underline{\nabla} T \quad (2.2)$$

where $\underline{q}$ is the heat flow vector, $\dot{e} = \frac{\partial e}{\partial t}$ the rate of specific volumetric enthalpy change, Q the rate of internally generated heat per unit volume, $\underline{k}$ a symmetric positive definite thermal conductivity matrix, T temperature, and t time. For isotropic materials

$$\underline{k} = k \underline{I} \quad (2.3)$$

where k is thermal conductivity, and $\underline{I}$ the identity matrix. The gradient operator $\underline{\nabla}$ is defined as

$$\underline{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (2.4)$$

where x , y , and z are Cartesian coordinates. Equation (2.2) is substituted into Equation (2.1) to yield the transient heat flow equation

$$-\underline{\nabla}^T (\underline{k} \underline{\nabla} T) + \dot{e} - Q = 0 \quad (2.5)$$

Specific volumetric enthalpy is by definition

$$e = \int_{T_0}^T c_\rho dT + \sum_i \ell_i \quad (2.6)$$

where T_0 is a reference temperature, usually zero, c specific heat, ρ density, and ℓ_i latent volumetric heat due to phase changes at various temperature levels. The time derivative of

$$\dot{e} = c_\rho \dot{T} \quad (2.7)$$

where $\dot{T} = \frac{\partial T}{\partial t}$ is rate of temperature change. Substitution of Equation (2.7) into Equation (2.5) yields the conventional form of the transient heat flow equation

$$-\nabla^T (k \nabla T) + c_\rho \dot{T} - Q = 0 \quad (2.8)$$

Nominal specific volumetric heat $\overline{c_\rho}$ will be defined by the equation

$$e = \overline{c_\rho} T \quad (2.9)$$

In Figure 2.1 specific volumetric enthalpy is plotted versus temperature for a material with latent heat indicated by a step ℓ in the curve. The tangential and secantial or nominal volumetric specific heats, c_ρ and $\overline{c_\rho}$, respectively, are then as shown in Figure 2.1. Note that at the temperature T_ℓ , where the enthalpy curve is stepped, the value of c_ρ is undefined while the value of $\overline{c_\rho}$ is always finite.

2.2 Initial and Boundary Conditions

Initial and boundary conditions must be specified in order to solve Equations (2.5) or (2.8). An initial condition is given by specifying the distribution of temperature in a body at a reference time zero. Boundary con-

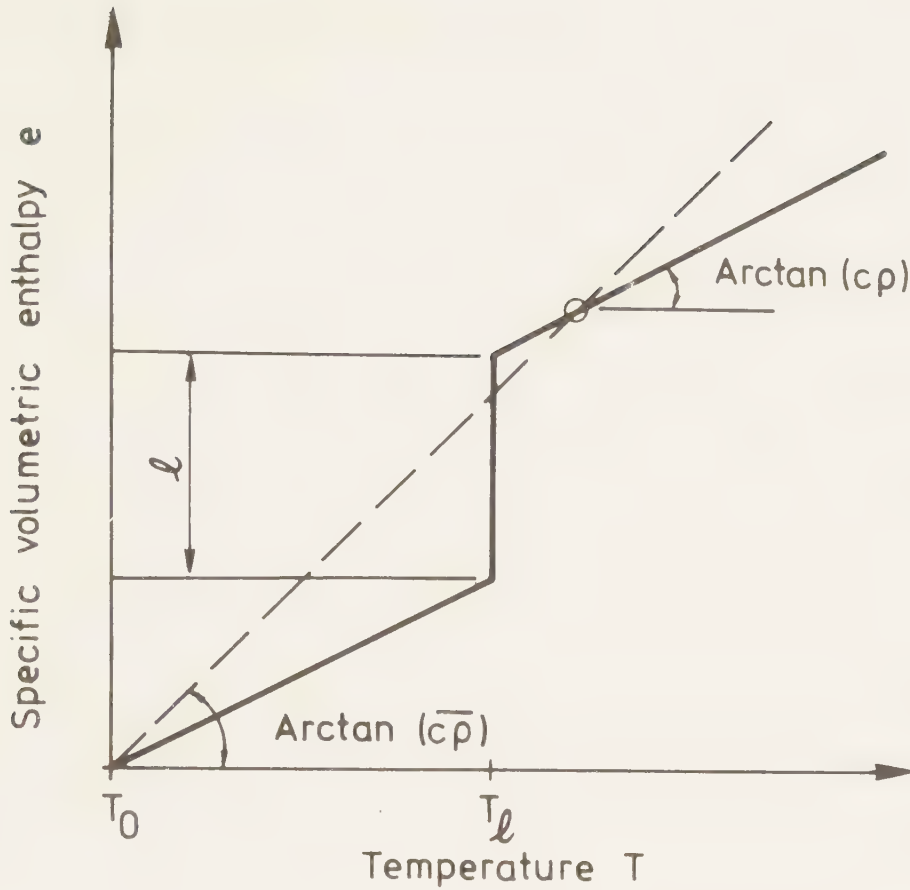


Figure 2.1. Definitions of specific volumetric heat

ditions are prescribed as temperature or heat flow on parts of the boundary ∂V_T and ∂V_q , respectively. The total boundary is then defined by

$$\partial V = \partial V_T + \partial V_q \quad (2.10)$$

Temperature on the boundary ∂V_T of a body is specified as

$$T = T(x, y, z, t) \quad (2.11)$$

Heat flow normal to a surface must satisfy the heat balance equation

$$q_n = \underline{n}^T \underline{q} = -\underline{n}^T \underline{k} \underline{\nabla} T \quad (2.12)$$

where $\underline{n}$ is the outward normal to the surface. Specified heat flow on ∂V_q therefore is

$$\hat{q}_n = -\underline{n}^T \underline{k} \underline{\nabla} T \quad (2.13)$$

where $\hat{q}_n$ is prescribed heat flow.

At free surfaces heat is transferred by convection and radiation. These phenomena are complex and difficult to model, but approximate formula can be used. Convection heat transfer is thus calculated as

$$\hat{q}_n^C = \beta (T_s - T_g)^\gamma \quad (2.14)$$

where $\hat{q}_n^C$ is the rate of heat transferred by convection, β and γ are the convection factor and power, respectively, and T_s and T_g are the surface and surrounding gas temperatures, respectively.

Radiation heat flux from a surface is approximated by

$$\hat{q}_n^R = \epsilon_r \sigma (\bar{T}_s^4 - \bar{T}_g^4) \quad (2.15)$$

where σ is the Stefan-Boltzmann constant, and $\bar{T}_s$ and $\bar{T}_g$ are absolute surface temperature and absolute surrounding gas temperature, respectively. Resultant emissivity ϵ_r varies with surface properties and geometric configuration. If the surface considered is small compared with a surrounding environment at uniform temperature T_g , resultant emissivity will be equal to surface emissivity ϵ_s [6]. When assessing radiation between flames and structures in fire engineering design, resultant emissivity is sometimes calculated assuming radiation between two infinitely long parallel planes [7]; thus,

$$\epsilon_r = \frac{1}{1/\epsilon_s + 1/\epsilon_g - 1} \quad (2.16)$$

where ϵ_g is appropriate gas or flame emissivity.

The total heat flux at a boundary is calculated by adding the contributions of convection and radiation:

$$\hat{q}_n = \hat{q}_n^c + \hat{q}_n^r \quad (2.17)$$

3. FINITE ELEMENT APPROXIMATION

3.1 Solution Techniques

Since analytical solutions of heat transfer problems are feasible only for linear applications with simple geometries and boundary conditions, a numerical method is used to solve the heat balance equation stated in Chapter 2 for temperature distribution in structural elements. The finite element method is used since it is general with respect to geometry, material properties, and boundary conditions. Nonlinear boundary conditions and the temperature dependence of material properties can be considered when the finite element method described in this chapter is used to analyze temperature distribution in fire-exposed structural elements.

3.2 Basic Approximations

In the finite element method of analysis a solid continuum is idealized by an assemblage of discrete elements. These elements may be of variable size and shape, and connected at a finite number of nodal points. The element boundaries are often linear, although if isoparametric elements are used, curved boundaries can be considered.

The temperature field within each element is approximated by a set of interpolation or shape functions N_i , chosen so as to define temperature uniquely within each finite element in terms of its nodal temperatures T_i . Temperature is thus approximated as

$$T = \sum_i N_i(x, y, z) T_i(t) = \underline{N} \underline{T} \quad (3.1)$$

The time differentiation of the temperature is

$$\dot{T} = \underline{N} \dot{\underline{T}} \quad (3.2)$$

Each shape function N_i is constructed so that it has the value 1 at node i and is zero at all other nodes. In elements adjacent to node i , N_i takes values less than unity, and in other elements it vanishes [8].

3.3 Matrix Equilibrium Equations for Transient Heat Conduction

The heat balance equilibrium equation for transient heat conduction in matrix form can be derived by various methods. The method of weighted residuals will be used here. Thus, Equation (2.9) is substituted into the heat balance equation, Equation (2.5); the resulting expression is multiplied by a weighting function v and integrated over the body [8]:

$$\int_V v (-\underline{\nabla}^T \underline{k} \underline{\nabla} T + \frac{\partial}{\partial t} (\overline{c\rho} T) - Q) dV = 0 \quad (3.3)$$

The first term is integrated by parts (Green's formula):

$$\begin{aligned} \int_V v (-\underline{\nabla}^T \underline{k} \underline{\nabla} T) dV &= - \int_{\partial V} v \underline{n}^T \underline{k} \underline{\nabla} T dS + \\ &+ \int_V (\underline{\nabla} v)^T \underline{k} \underline{\nabla} T dV \end{aligned} \quad (3.4)$$

where $\underline{n}$ is the outward normal to the boundary ∂V . A set of weighting functions v_i equal to the shape functions N_i (the Galerkin method) is then chosen, i.e.

$$v_i = N_i \quad (3.5)$$

Equations (3.1, 3.2, 3.4, 3.5) are substituted into Equation (3.3), yielding the matrix heat balance equation

$$\begin{aligned} \int_V \left[(\underline{\nabla} \underline{N})^T \underline{k} \underline{\nabla} \underline{N} dV \right] \underline{T} + \frac{\partial}{\partial t} \left[\int_V \underline{N}^T \overline{c\rho} \underline{N} dV \underline{T} \right] = \\ = \int_V \underline{N}^T Q dV + \int_{\partial V} \underline{N}^T \underline{n}^T \underline{k} \underline{\nabla} T dS \end{aligned} \quad (3.6)$$

or

$$\underline{F}_T + \frac{\partial}{\partial t}(\underline{E}) = \underline{F}_Q + \underline{F}_q \quad (3.7)$$

where $\underline{F}_T$, $\underline{E}$, $\underline{F}_Q$ and $\underline{F}_q$ are vectors of nodal heat flow due to conduction, enthalpy or heat stored in elements adjacent to nodes, rate of internally generated heat per unit volume, and rate of heat flow supplied at the boundary, respectively. The vector of internal heat flow due to conduction is

$$\underline{F}_T = \underline{K} \underline{T} \quad (3.8)$$

where $\underline{K}$ is the heat conductivity matrix.

Equation (3.8) is substituted into Equation (3.7) to yield

$$\underline{K} \underline{T} + \frac{\partial}{\partial t}(\underline{E}) = \underline{F} \quad (3.9)$$

where

$$\underline{F} = \underline{F}_Q + \underline{F}_q \quad (3.10)$$

The nodal enthalpy vector is

$$\underline{E} = \underline{C} \underline{T} \quad (3.11)$$

where $\underline{C}$ is the nominal heat capacity matrix. This expression is substituted into Equation (3.9)

$$\underline{K} \underline{T} + \frac{\partial}{\partial t}(\underline{C} \underline{T}) = \underline{F} \quad (3.12)$$

Alternatively, the heat balance equation can be expressed in terms of nodal enthalpy rather than in terms of temperature

$$\underline{K}^* \underline{E} + \frac{\partial}{\partial t}(\underline{E}) = \underline{F} \quad (3.13)$$

where

$$\underline{K}^* = \underline{K} \underline{C}^{-1} \quad (3.14)$$

The integrals in Equation (3.6) are evaluated over all elements m and boundary elements ∂m . Thus

$$\underline{K} = \sum_m \underline{K}^m \quad (3.15)$$

$$\underline{C} = \sum_m \underline{C}^m \quad (3.16)$$

$$\underline{F} = \sum_m \underline{F}_Q^m + \sum_{\partial m} \underline{F}_q^{\partial m} \quad (3.17)$$

where

$$K_{ij}^m = \int_{V^m} (\underline{\nabla} N_i)^T \underline{k} (\underline{\nabla} N_j) dV \quad (3.18)$$

$$C_{ij}^m = \int_{V^m} N_i \overline{c_\rho} N_j dV \quad (3.19)$$

$$F_{Qi}^m = \int_{V^m} N_i Q dV \quad (3.20)$$

and

$$F_{qi}^{\partial m} = \int_{\partial V^m} N_i \underline{n}^T \underline{k} \underline{\nabla} T dS \quad (3.21)$$

V^m and ∂V^m are element volumes and boundary element surfaces, respectively.

The integrals of Equations (3.18-3.21) are often solved numerically by Gaussian quadrature. Explicit expressions can be derived for simple rectangular two dimensional elements as used in TASEF-2, as will be shown in the following sections.

3.3.1 Conductivity Matrix

In this section the element conductivity matrix $\underline{K}^m$ for the simple two-dimensional rectangular element used in

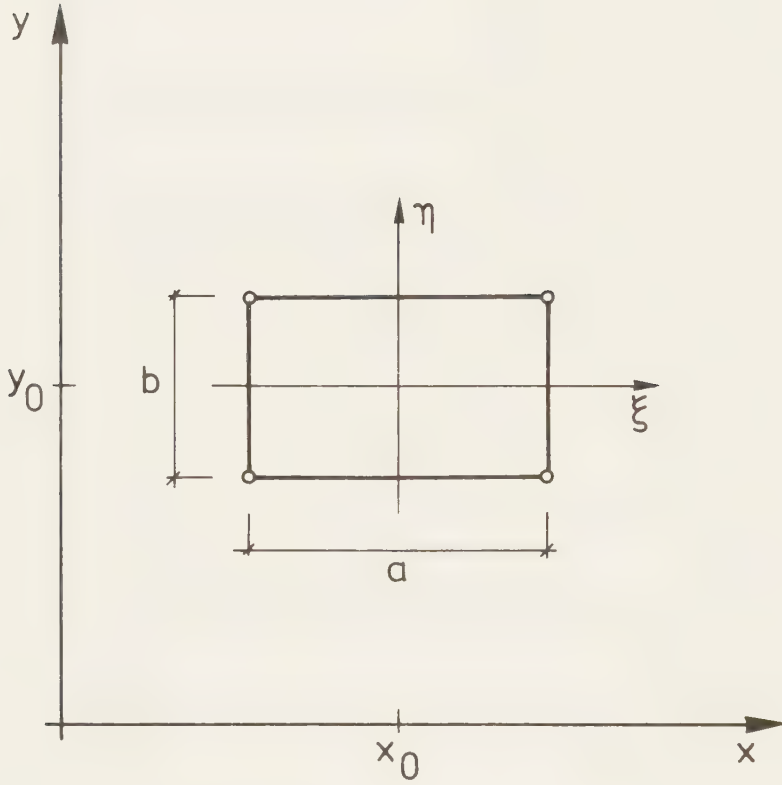


Figure 3.1. Rectangular finite element

program TASEF-2 will be derived. Consider the rectangular element with sides parallel with the axes and of lengths a and b as shown in Figure 3.1. Make the variable substitutions

$$\xi = (x - x_0)/a \quad (3.22)$$

and

$$\eta = (y - y_0)/b \quad (3.23)$$

where ξ and η are dimensionless coordinates in a local system. A set of allowable shape functions is then

$$N_i = (1 + \xi\xi_i)(1 + \eta\eta_i)/4 \quad (3.24)$$

where i takes values from 1 to 4, and

$$\underline{\nabla} N_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{1}{a} \frac{\partial N_i}{\partial \xi} \\ \frac{1}{b} \frac{\partial N_i}{\partial \eta} \end{bmatrix} = 1/4 \begin{bmatrix} \frac{\xi_i}{a} (1 + \eta \eta_i) \\ \frac{\eta_i}{b} (1 + \xi \xi_i) \end{bmatrix} \quad (3.25)$$

Equation (3.25) is substituted into Equation (3.18) and constant thickness d and conductivity k are assumed for the element. Thus the local element conductivity matrix $\underline{K}^m$ is, after evaluation of simple integrals,

$$\underline{K}^m = \frac{1}{3} \frac{kd}{ab} \begin{bmatrix} a^2 + b^2 & -b^2 + \frac{a^2}{2} & -\frac{b^2}{2} - \frac{a^2}{2} & \frac{b^2}{2} - a^2 \\ & a^2 + b^2 & \frac{b^2}{2} - a^2 & -\frac{b^2}{2} - \frac{a^2}{2} \\ & & a^2 + b^2 & -b^2 + \frac{a^2}{2} \\ \text{sym.} & & & a^2 + b^2 \end{bmatrix} \quad (3.26)$$

If conductivity k for a particular application varies with temperature, k at average nodal temperature is used in the calculation.

3.3.2 Heat Capacity and Volume Matrices

The computation of the element heat capacity matrix $\underline{C}^m$ as given in Equation (3.19) results in a fully populated matrix identical in form to the element conductivity matrix $\underline{K}^m$. The assembled heat capacity matrix $\underline{C}$ is symmetric, positive-definite, and has the same nonzero structure as the system conductivity matrix $\underline{K}$. The element heat capacity matrix $\underline{C}^m$ can, however, be approximated by a lumped diagonal matrix with no loss of accuracy. The lumping eliminates the coupling between the time rate-of-change of temperature at adjacent nodes and results in a diagonal heat capacity matrix $\underline{C}$. Such an approximation facilitates solution of the heat balance equation as will be shown in Section 3.4.

The lumped element heat capacity matrix C_{ii}^m is formed in TASEF-2 as:

$$C_{ii}^m = \overline{c\rho}^m(T_i) W_{ii}^m \quad (3.27)$$

where $\overline{c\rho}^m(T_i)$ is nominal specific volumetric heat capacity at nodal temperature T_i and W_{ii}^m the volume of element m associated with node i . For rectangular 4-node elements the volume associated with each node is a quarter of an element. If all elements connected at a node i are of the same material, the lumped heat capacity matrix can be stated as:

$$C_{ii} = \overline{c\rho}(T_i) W_{ii} \quad (3.28)$$

where

$$W_{ii} = \sum_m W_{ii}^m \quad (3.29)$$

defines the global diagonal volume matrix.

3.3.3 Internally Generated Heat

Internally generated heat is calculated elementwise using the volume matrix $\underline{W}$. Thus for a node i

$$F_{Qi} = \sum_m Q_i^m W_{ii}^m \quad (3.30)$$

where Q_i^m is the rate of heat generated per unit volume at node i in element m and W_{ii}^m is the volume adjacent to node i of element m . In TASEF-2 the rate of internally generated heat is input as a function of temperature.

3.3.4 Boundary Heat Flow

Either heat flow F_{qi} or temperature T_i are prescribed for all nodes i . On that part ∂V_q of the boundary where heat

flow is prescribed, nodal heat flow is calculated by substituting Equation (2.13) into Equation (3.21):

$$F_{qi}^{\partial m} = - \int_{\partial V_q^m} N_i \hat{q}_n dS \quad (3.31)$$

The shape functions N_i are linear along the boundaries. Thus for a boundary element ∂m with lengths s and thickness d as shown in Figure 3.2 the nodal heat flow to an adjacent node i is

$$F_{qi}^{\partial m} = -1/6 sd(2\hat{q}_{ni}^{\partial m} + \hat{q}_{nj}^{\partial m}) \quad (3.32)$$

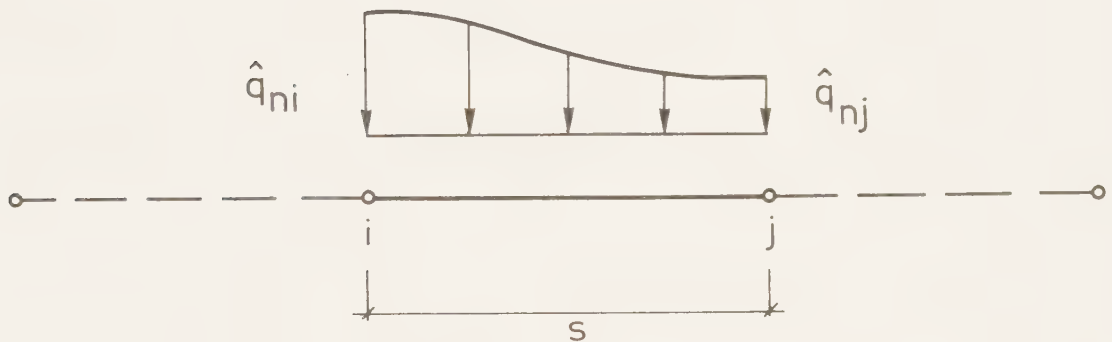


Figure 3.2. Heat flow to a boundary element

Equations (2.14 and 2.15) are then used to yield

$$\hat{q}_{ni}^{\partial m} = - \left[\epsilon_r \sigma (\bar{T}_{gi}^4 - \bar{T}_i^4) + \beta (\bar{T}_{gi} - \bar{T}_i)^\gamma \right] \quad (3.33)$$

where $\bar{T}_{gi}$ and $\bar{T}_i$ are absolute gas and surface temperature at node i , respectively; ϵ_r is resultant emissivity, and β and γ are convection factor and power, respectively, for a boundary element ∂m . Equation (3.33) is substituted into Equation (3.32) to yield

$$F_{qi}^{\partial m} = B_{rii}^{\partial m} T_{ri} + B_{rij}^{\partial m} T_{rj} + B_{cii}^{\partial m} T_{ci} + B_{cij}^{\partial m} T_{cj} \quad (3.34)$$

where i and j are nodes adjacent to a boundary element ∂m and

$$B_{rii}^{\partial m} = \frac{1}{3} s d \epsilon_r \sigma \quad (3.35)$$

$$B_{rij}^{\partial m} = \frac{1}{2} B_{rii}^{\partial m} \quad (3.36)$$

$$B_{cii}^{\partial m} = \frac{1}{3} s d \beta \quad (3.37)$$

$$B_{cij}^{\partial m} = \frac{1}{2} B_{cii}^{\partial m} \quad (3.38)$$

$$T_{ri} = \bar{T}_{gi}^4 - \bar{T}_i^4 \quad (3.39)$$

and

$$T_{ci} = (\bar{T}_{gi} - \bar{T}_i)^\gamma \quad (3.40)$$

External heat flow to all boundary nodes is assembled in matrix form to

$$\underline{F}_q = \underline{B}_r \underline{T}_r + \underline{B}_c \underline{T}_c \quad (3.41)$$

where $\underline{T}_r$ and $\underline{T}_c$ are vectors of modified nodal temperature as defined by Equations (3.39 and 3.40), respectively, and $\underline{B}_r$ and $\underline{B}_c$ are boundary radiation and convection matrices, respectively, where

$$B_{rij} = \sum_{\partial m} B_{rij}^{\partial m} \quad (3.42)$$

and

$$B_{cij} = \sum_{\partial m} B_{cij}^{\partial m} \quad (3.43)$$

Summation need be carried out only for the two boundary elements adjacent to a node i as only these contribute to the external heat flow to that node.

In TASEF-2 boundary nodes must be input sequentially around the boundary. The boundary matrices $\underline{B}_r$ and $\underline{B}_c$ then become tri-diagonal, i.e. only elements in the diagonal and adjacent to the diagonal have nonzero values, and since they are symmetric only two column matrices need be stored. The boundary matrices will remain constant and need be established only once when emissivity ϵ_r and convection factor β are assumed constant.

3.4 Time Integration

The heat flow equilibrium equation in matrix form may be solved by directly integrating the coupled differential equation step-by-step. If nodal enthalpy and external heat flow are assumed to vary linearly within each time step, Equation (3.13) can be approximated as

$$\begin{aligned} \theta \underline{K}_{t+\Delta t}^* \underline{E}_{t+\Delta t} + (1-\theta) \underline{K}_t^* \underline{E}_t + (\underline{E}_{t+\Delta t} - \underline{E}_t) / \Delta t = \\ = \theta \underline{F}_{t+\Delta t} + (1-\theta) \underline{F}_t \end{aligned} \quad (3.44)$$

where the indices indicates time, and where θ is an arbitrary parameter in the range

$$0 \leq \theta \leq 1 \quad (3.45)$$

If different values are assigned to θ various time integration schemes are defined. Thus for $\theta = 0$, 0.5, and 1, the wellknown forward-, mid-, and backward-difference methods, respectively, are obtained. While for linear problems the latter two methods are unconditionally stable, i.e. for any time increment Δt used solutions will not diverge, the forward-difference method will converge only if the time-increment Δt is less than a critical value Δt_{cr} . The value of this critical time increment depends on element size, material properties, and boundary conditions. If $\underline{C}$ is a diagonal (lumped) matrix the solution for $\underline{T}_{t+\Delta t}$ is straight forward; each value can be computed directly from its precursor without the need to solve

simultaneous equations. Thus the forward-difference method is explicit while the mid- and backward-difference methods are implicit and require an equation system to be solved at each time step. Although such solutions can be very costly for nonlinear problems, implicit methods are often used because they are unconditionally stable with respect to length of time increment.

Time increments are, however, also limited by the requirement that variations in boundary conditions and material properties be adequately followed. Therefore in many problems in fire engineering, short time increments must be used even if implicit methods are employed; the magnitude of the critical time steps for the explicit Euler method is thus often the same as that required to follow changes in boundary conditions. Since during each time-step explicit methods require less computation, the forward-difference method becomes favourable. In the following section it will also be shown that the Euler method is particularly advantageous when specific heat for a material varies with temperature or when energy-consuming phase changes occur.

3.4.1 Forward Differences

For $\theta = 0$ in Equation (3.44) the explicit forward difference formula is

$$\underline{E}_{t+\Delta t} = \underline{E}_t + (\underline{F}_t - \underline{K}_t^* \underline{E}_t) \Delta t \quad (3.46)$$

or after substitution of Equation (3.11) and (3.14)

$$\underline{E}_{t+\Delta t} = \underline{E}_t + (\underline{F}_t - \underline{K}_t \underline{T}_t) \Delta t \quad (3.47)$$

Equation (3.11) is then used to obtain temperature at a node i :

$$T_{i,t+\Delta t} = C_{ii}^{-1}(T_{i,t+\Delta t})E_{i,t+\Delta t} \quad (3.48)$$

If C_{ii} varies with temperature the exact solution of Equation (3.48) is obtained by iteration. However, if all elements around a node i are of the same material the specific volumetric enthalpy is calculated as

$$e_{i,t+\Delta t} = E_{i,t+\Delta t}/W_{ii} \quad (3.49)$$

and $T_{i,t+\Delta t}$ is obtained by using the temperature-specific volumetric enthalpy relation as shown in Figure 3.3. The latter method is computationally very fast and is therefore used when ever possible in TASEF-2. For nodes at interfaces between elements of different materials, the following iteration formula is used to calculate temperature

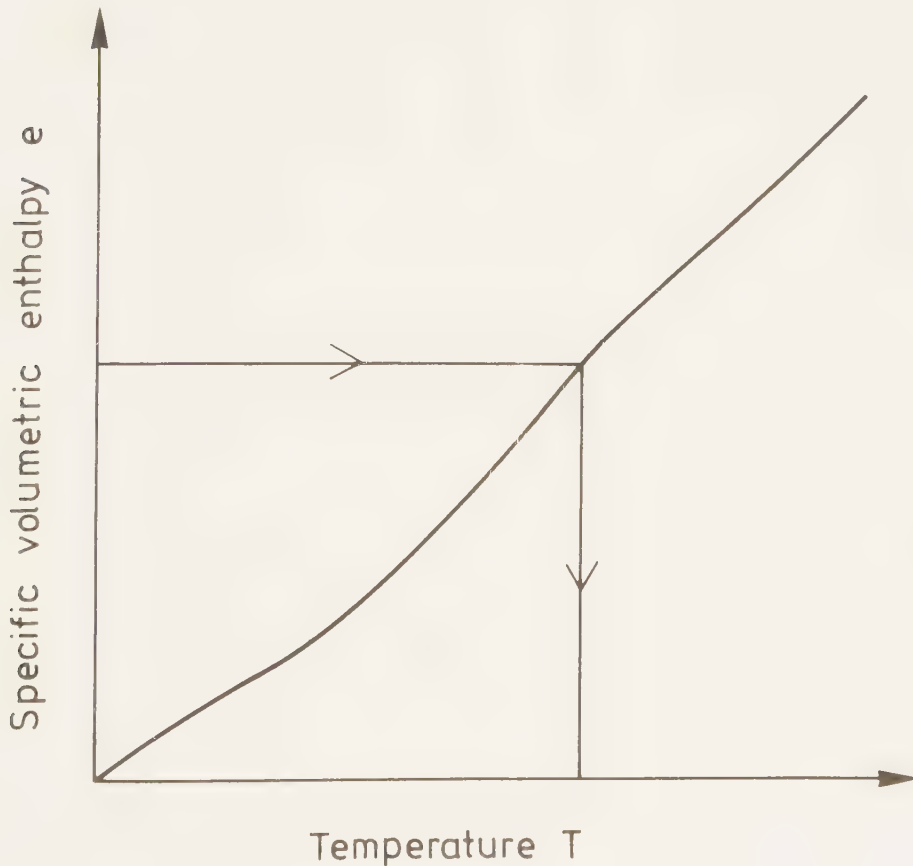


Figure 3.3. Translation of specific volumetric enthalpy into temperature

$$T_{i,t+\Delta t}^{j+1} = C_{ii}^{-1} (T_{i,t+\Delta t}^j) E_{i,t+\Delta t} \quad (3.50)$$

where j refers to iteration steps. For the first iteration step temperature from the previous time step is assumed. Iteration terminates when the difference between the nodal temperature from two successive iterations is less than a permissible value δ expressed as

$$\frac{T_{i,t+\Delta t}^{j+1} - T_{i,t+\Delta t}^j}{T_{i,t+\Delta t}^{j+1} + T_{i,t+\Delta t}^j} < \delta/2 \quad (3.51)$$

δ is in TASEF-2 set equal 1%. Normally, convergence is achieved in a small number of iteration steps.

3.4.2 Critical Time Increment

To derive a simple expression by which the critical time increment Δt_{cr} can be estimated, the first steps in a modal solution of the heat flow equilibrium equation are shown below. If the nominal heat capacity matrix $\underline{C}$ is assumed to be time independent, Equation (3.12) is

$$\underline{C} \frac{\partial}{\partial t} \underline{T} + \underline{K} \underline{T} = \underline{F} \quad (3.52)$$

At any time step, the righthand side of Equation (3.51) can be linearized at current temperature; thus, matrix $\underline{K}_F$ is defined by

$$K_{Fij} = \frac{dF_i}{dT_j} \quad (3.53)$$

where i and j denote rows and columns. Thus the homogeneous part of Equation (3.52) is

$$\underline{C} \frac{\partial}{\partial t} \underline{T} + \bar{\underline{K}} \underline{T} = \underline{0} \quad (3.54)$$

where

$$\bar{\underline{K}} = \underline{K} - \underline{K}_F \quad (3.55)$$

In case of homogeneous boundary conditions, solutions of Equation (3.54) have the form

$$\underline{T} = e^{-\lambda t} \underline{\phi} \quad (3.56)$$

where $\underline{\phi}$ is a vector independent of time t . Multiply by the inverse of the diagonal matrix $\underline{C}$:

$$-\lambda e^{-\lambda t} \underline{\phi} + \underline{C}^{-1} \underline{\bar{K}} e^{-\lambda t} \underline{\phi} = 0 \quad (3.57)$$

Because $e^{-\lambda t}$ can never be equal zero, the eigenvalue problem

$$(\underline{C}^{-1} \underline{\bar{K}}) \underline{\phi} = \lambda \underline{\phi} \quad (3.58)$$

arises. Equation (3.58) is an n :th order equation where n is the number of temperature degrees of freedom in the system. There are n solutions of eigenvalues (thermal frequencies) $\lambda_1, \lambda_2 \dots \lambda_n$ with corresponding eigenvectors (thermal modes) $\phi_1, \phi_2 \dots \phi_n$.

The critical time increment for a forward difference scheme is now obtained [8]

$$\Delta t_{cr} = \frac{2}{\lambda_{max}} \quad (3.59)$$

where λ_{max} is the maximum eigenvalue.

Exact calculation of λ_{max} at every time step is very time consuming. The Gerschgorin's theorem [9], however, states that the maximum eigenvalue of a matrix with elements a_{ij} is

$$\lambda \leq \max_i (a_{ii} + \sum_{j \neq i} |a_{ij}|) \quad (3.60)$$

where i and j are rows and columns, respectively. The

diagonal elements of the heat conduction matrix $\underline{K}$ are equal to the negative sum of the off-diagonal elements of the corresponding row, i.e.

$$K_{ii} = -\sum_j K_{ij} \quad (3.61)$$

Thus the maximum eigenvalue of Equation (3.58) is

$$\lambda_{\max} \leq \max_i \left[C_{ii}^{-1} (2K_{ii} + \sum_j K_{Fij}) \right] \quad (3.62)$$

and an upper limit to the critical time increment is

$$\Delta t_{cr} = \min_i \left[\frac{C_{ii}}{K_{ii} + \frac{1}{2} \sum_j K_{Fij}} \right] \quad (3.63)$$

This approximation is used in TASEF-2 to update the critical time increment at each time step; time increments are thus continually adjusted to account for current conditions.

In Equation (3.63) it is implicit that for nodes for which the ratio of heat capacity to thermal conductance to adjacent nodes is small, the critical time increment will be very small. When possible without jeopardizing accuracy, thermal resistance between such nodes can then be neglected; the temperature of these nodes is set to the same value. All terms for these coupled nodes are combined. The resulting denominator in Equation (3.63) is reduced while C_{ii} is increased; the resulting critical time step for this region is thus substantially increased. When calculating temperature in fire-exposed steel structures, for example, the difference in temperature between opposite sides of steel sheets will in most cases be negligible. Corresponding nodes can therefore be coupled without losing accuracy (see Example II and III in Section 5). At boundaries for which the heat transfer coefficient is high, short time increments may be avoided by prescribing surface temperature instead of heat transfer. This approximation is particularly useful when analyzing heat transfer in light insulating materials.

4. COMPUTER PROGRAM

The computer program TASEF-2 (Temperature Anal^ysⁱs of Structures Exposed to Fire - Two Dimensional Version) is developed for the analysis of thermal response of a variety of structures exposed to fire. It is coded on the basis of the theory presented in previous sections of this report. All subroutines are coded in Fortran V, while the main program is coded in NuAlgol in order to permit dynamic allocation of arrays. As all storage is in core, the number of nodes and elements in a structure is limited by available computer memory.

Input of geometric data to the current version of TASEF-2 has been automated. A structure is generated from a base rectangle with two sides that coincide with the x- and y-axes, and two at maximum x- and y-coordinates. A mesh is then generated by lines either at specified distances or at prescribed coordinates. Rectangular subregions either with elements of different material than that of the main region, or fictitious elements in voids or cut outs from the base rectangle, are defined in the input by their minimum and maximum x- and y-coordinates. Any structure that can be assembled of rectangular elements is therefore easily generated.

The material properties conductivity and specific volumetric enthalpy are assumed to vary piecewise linearly with temperature, and are input for each region as a number of temperature property-value pairs. As the conductivity of heated concrete in the cooling phase remains approximately as at maximum temperature, the user can specify that, for appropriate regions, conductivity in the cooling phase is to be calculated as a function of maximum instead of current temperature.

The critical time increment for nodes close to each other or separated by a material with high thermal diffusivity will be very short (see Section 3.4.2). Such nodes may be coupled to other adjacent nodes, i.e. their temperature will be prescribed to be equal. Errors thus introduced are negligible if the exact temperature at the coupled nodes differs little.

Nodes with common properties can be grouped to facilitate input and computation. Such groups may consist of nodes at boundaries with prescribed temperature or heat transfer conditions. Node groups are also used to define voids where heat transfer by convection and radiation occur. Emissivity and convection factor and power are assigned to node groups, where appropriate.

Heat exchange by convection and radiation between enclosure surfaces in structures with voids may be considered. The procedure is fully described in [10]. View factors between surfaces defined by the nodes on the enclosure surfaces are calculated automatically by the program. Convection is computed assuming that no exchange of enclosed air occurs and that heat stored in the air is negligible for the heat balance of the surrounding solid. Portions of enclosure surfaces are assigned heat transfer properties by using several node groups to define each void.

The temperature of boundary nodes or of the surrounding gas is defined as a constant ambient temperature or a time-dependent fire temperature. A fire temperature history is specified by a number of points on a time-temperature curve. Temperature between these points is obtained by linear interpolation. If the time-temperature relation specified for the ISO 834 standard fire resistance test [11] is assumed, the fire temperature T_f may instead be calculated as

$$T_f = T_o + 345 \log_{10}(430t + 1) \quad (4.1a)$$

for $t \leq t_u$, when T_o is ambient temperature, and t and t_u are time and duration of heating phase, respectively, in hours. In the cooling phase the fire temperature decreases at the following rates:

$$\begin{array}{llll} 625 & ^\circ\text{C/h} & \text{if } t_u \leq 0.5 & \text{h} \\ 250(3-t_u) & ^\circ\text{C/h} & \text{if } 0.5 \leq t_u \leq 2 & \text{h} \\ 250 & ^\circ\text{C/h} & \text{if } t_u > 2 & \text{h} \end{array} \quad (4.1b)$$

The forward difference time integration scheme described in Section 3.4 is used. The conductivity matrix (Section 3.3.1) is symmetric and banded, and therefore only the lower half band including the diagonal of the matrix is formed. The heat capacity and volume matrices (Section 3.3.2) are diagonal (lumped) and are therefore stored as vectors. The conductivity matrix is updated either at each time step or at intervals specified in the input. Computer time is saved if updating can be avoided without sacrificing accuracy.

At each time step the length of the critical time increment is computed as described in Section 3.4.2 and a time increment is obtained by multiplying by a time increment factor specified by the user, usually in the range 0.75-1.0. Values greater than this range may cause numerical instability, while smaller values will prolong computation without necessarily increasing accuracy. If the critical time increment is very long in relation to the rate of change of boundary temperature, the user may specify an upper limit to the time increment.

Nodal temperature is printed at specified times. Maximum nodal temperatures obtained during an analysis are currently stored in a vector and are used in calculations of conductivity for certain elements as described above. When a nodal temperature begins to decrease, node number, maximum

temperature, and time of occurrence are printed. Finally, when the analysis is terminated at a time specified in the input, maximum nodal temperatures are printed.

A summary of the solution technique and detailed input instructions and a complete listing of the program are given in Appendices A and B, respectively.

5. EXAMPLES

Three examples were solved to demonstrate the use and verify the accuracy of TASEF-2. The solution of the first problem is compared to an analytical solution and results from the others to experimental data. Input cards for the examples are listed in Appendix C.

5.1 Example I - Square Plate Subjected to Heat Transfer from Surrounding Gas

A square plate with side lengths 2ℓ as shown in Figure 5.1a initially at uniform temperature T_o is suddenly subjected to an environment of uniform gas temperature T_g . Heat transfer q at the boundary to the body surface is

$$q = h(T_g - T_s)$$

where T_s is surface temperature and h a heat transfer coefficient assumed constant. Conductivity k , density ρ , and specific heat capacity c are assumed constant. The following dimensionless parameters are introduced for convenience:

$$\theta = (T - T_g)/(T_o - T_g)$$

$$Fo = at/\ell^2$$

$$Bi = h\ell/k$$

where the thermal diffusivity a is

$$a = k/c\rho$$

and Fo and Bi are the Fourier and Biot numbers, respectively. By separation of variables, analytical product

solutions as infinite sums are obtained to this problem [12]. The temperature at the center of the plate was calculated analytically and numerically by the program TASEF-2 for $Bi = 1$.

As the problem is symmetrical only one quadrant need be analyzed. Numerical solutions were obtained using a coarse mesh of 4 elements and a fine mesh of 16 elements as shown in Figure 5.1b. By assigning a time increment factor equal to one, time steps equal to $0.0667 a/l^2$ and $0.0200 a/l^2$ were calculated for the coarse and fine meshes, respectively.

Numerical results obtained with TASEF-2 and the exact analytical solution are given in Table 5.1. Errors in the numerical solutions are small even for the coarse mesh with only 4 elements.

5.2 Steel Beams Embedded in Concrete

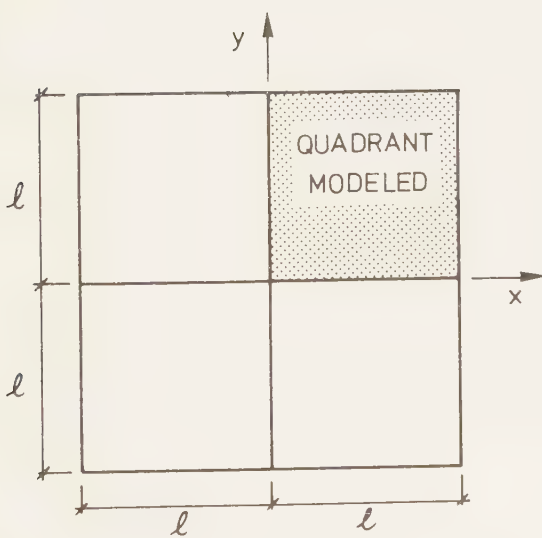
5.2.1 Material Properties and Boundary Conditions

A wide-flange I-beam and a box girder of steel embedded in normal concrete, as shown in Figures 5.2 and 5.3, were exposed on one side to a model fire that approximately corresponded to the ISO 834 standard time-temperature curve in a test furnace. Steel and concrete temperatures were measured at several points over the cross section and compared to temperature predicted by the program TASEF-2.

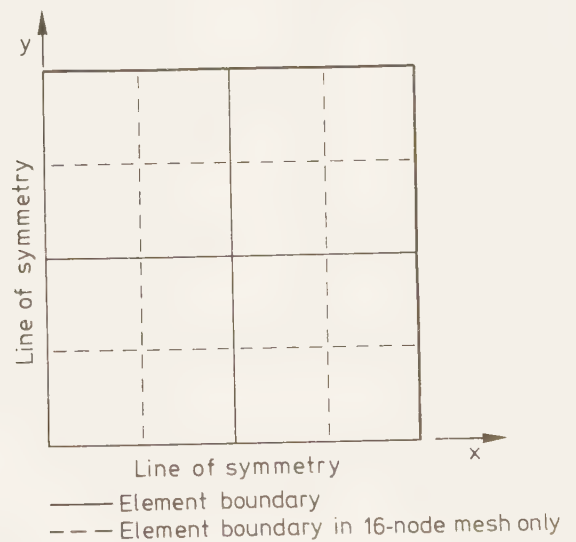
Conductivity and specific volumetric heat of steel were assumed to vary with temperature as shown in Figures 5.4 and 5.5 [13]. Latent heat due to phase changes at temperature around 725°C is considered. The thermal properties of concrete vary considerably with type of mix, moisture content, curing, age, etc. The assumed temperature-

Table 5.1. Comparison of analytically [11] and numerically calculated dimensionless temperature θ at center of square plate exposed to heat transfer from surrounding gas, $Bi=1$

Dimensionless time Fo	Exact solution	Coarse mesh 4 elements $\Delta t=0.0667 \text{ at}/\ell^2$		Fine mesh 16 elements $\Delta t=0.0200 \text{ at}/\ell^2$	
		Numerical solution	Error	Numerical solution	Error
0.1	0.9864			0.993	-0.007
0.2	0.9038	0.925	-0.021	0.909	-0.005
0.4	0.6902	0.688	0.002	0.690	-
0.6	0.5147	0.505	0.009	0.512	0.003
0.8	0.3827	0.370	0.012	0.379	0.004
1.0	0.2845	0.271	0.013	0.281	0.004
Number of time steps		15		50	



(a) Square plate



(b) Finite element meshes for square quadrant

Figure 5.1. Finite element model for calculating heat transfer in a square plate

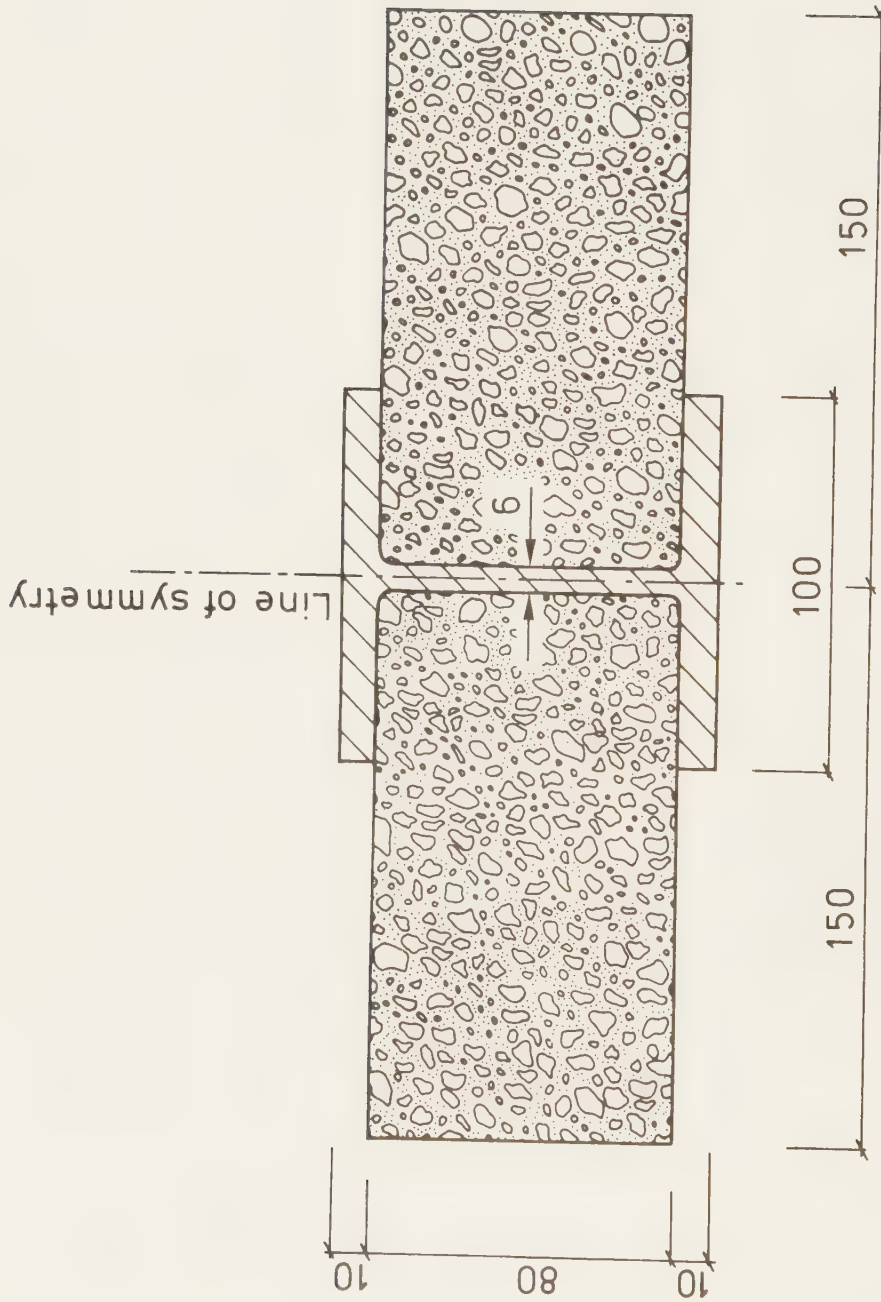


Figure 5.2. Wide-flange I-beam (HE100B) embedded in a concrete slab.
The vertical sides were insulated during furnace test.
Dimensions in mm

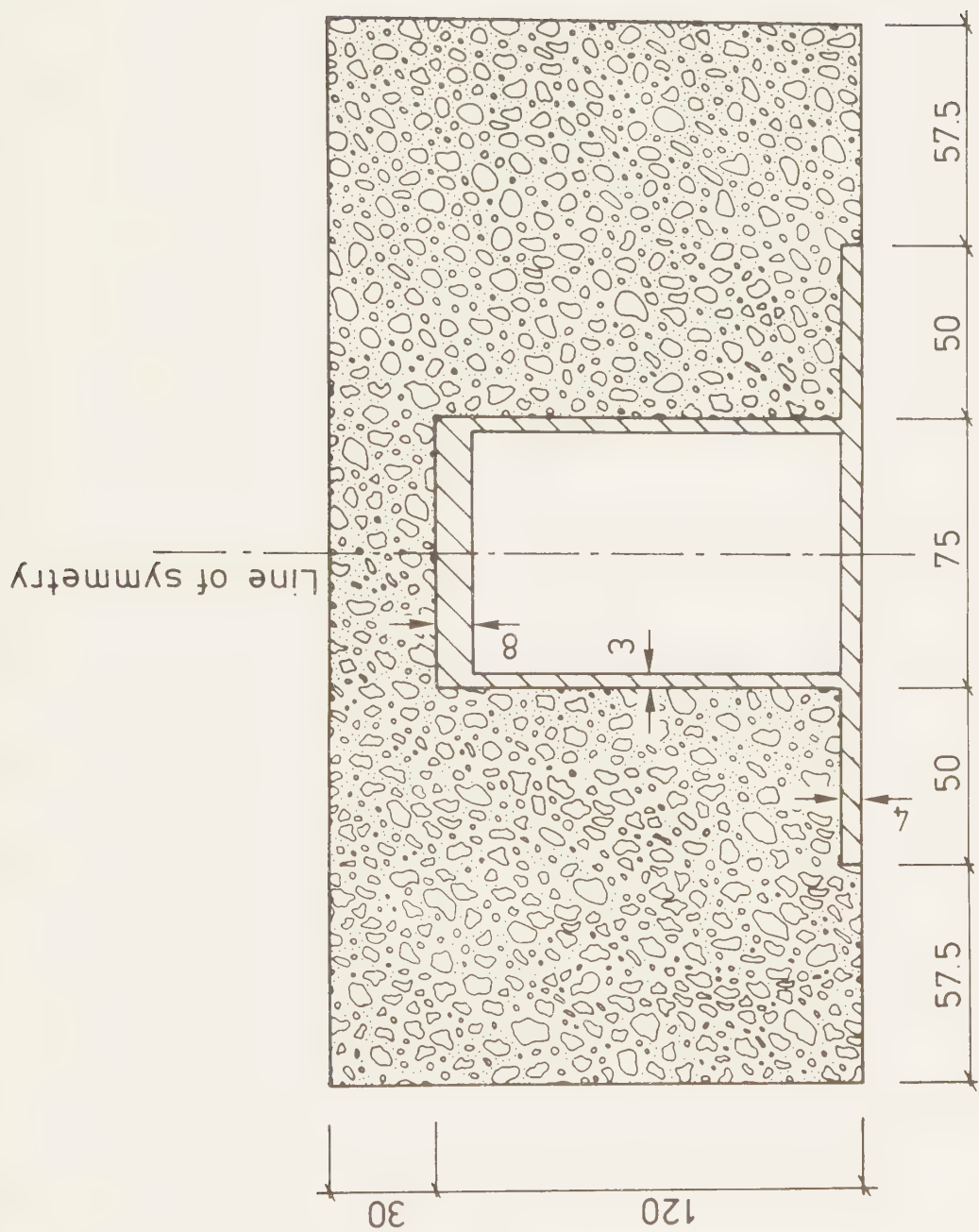


Figure 5.3. Box girder embedded in a concrete slab. The vertical concrete sides were insulated during furnace test.
Dimensions in mm

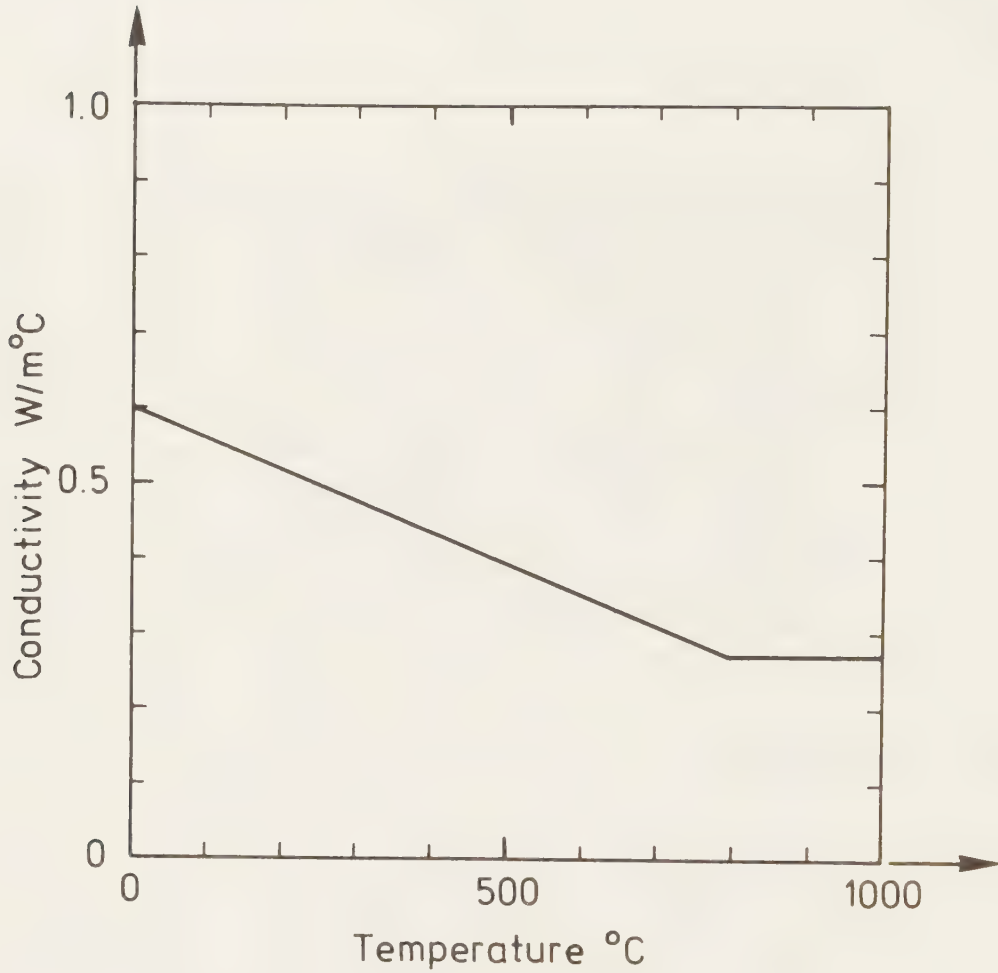


Figure 5.4. Conductivity of steel

conductivity relation (Figure 5.6) was that measured by the Stålhane Pyk method for the type of concrete used in the test [14]. The assumed variation of specific volumetric enthalpy with temperature is based on measurements on dry concrete [15]. Enthalpy corresponding to heating and evaporation of moisture is then calculated and added. Thus the specific enthalpy e for concrete with a moisture content u is

$$e(T) = e^C(T) + u e^W(T)$$

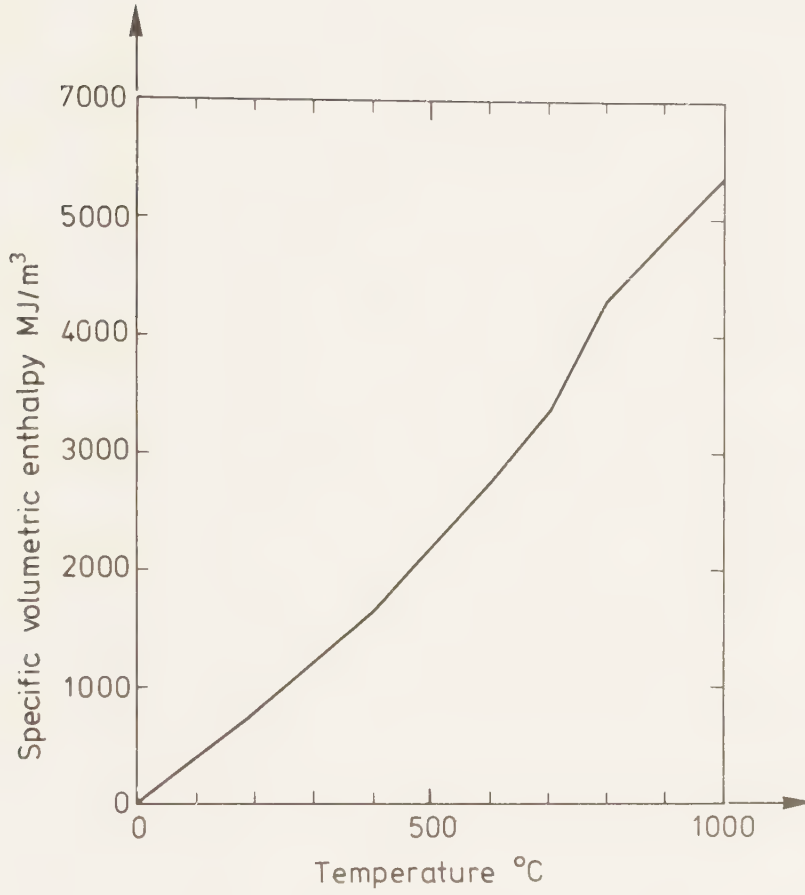


Figure 5.5. Specific volumetric enthalpy of steel

where e^c and e^w are specific enthalpy for dry concrete and water, respectively. If the water is assumed to evaporate linearly in the temperature range of T_1 to T_2 :

$$e^w(T) = c^w T \quad \text{for } T < T_1$$

$$e^w(T_2) = e^w(T_1) + \frac{1}{2} c^w (T_2 - T_1) + a^w \quad \text{for } T = T_2$$

$$e^w(T) = e^w(T_2) \quad \text{for } T > T_2$$

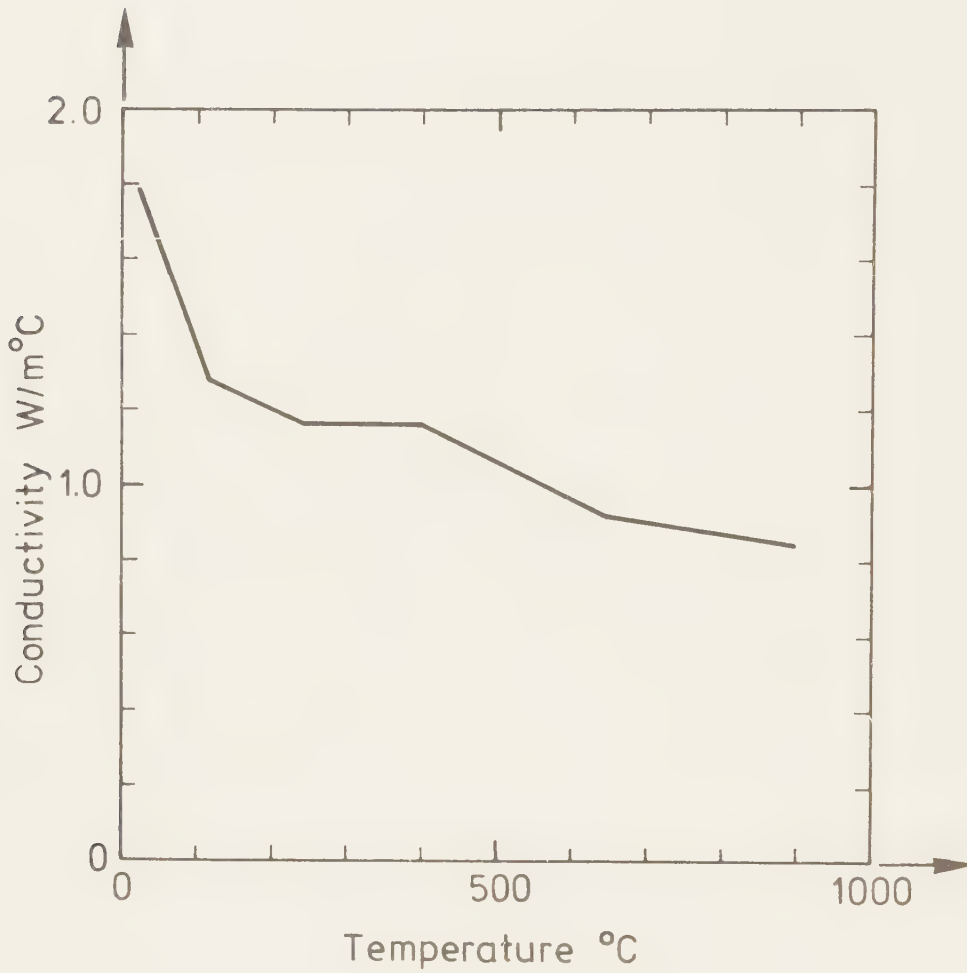


Figure 5.6. Thermal conductivity of concrete

where c_w^w and a_w^w are the specific heat and heat of evaporation, respectively, of water. The specific volumetric enthalpy e_v is then obtained by multiplying by the density ρ^c of concrete:

$$e_v(T) = \rho^c e(T)$$

As the test specimens were cured in an environment of 40% relative humidity for a month, a moisture content

of 1.5% by weight was assumed [16]. The specific volumetric enthalpy for dry and moist concrete, where moisture is assumed to evaporate in the temperature range of 100 to 115°C, is plotted versus temperature in Figure 5.7.

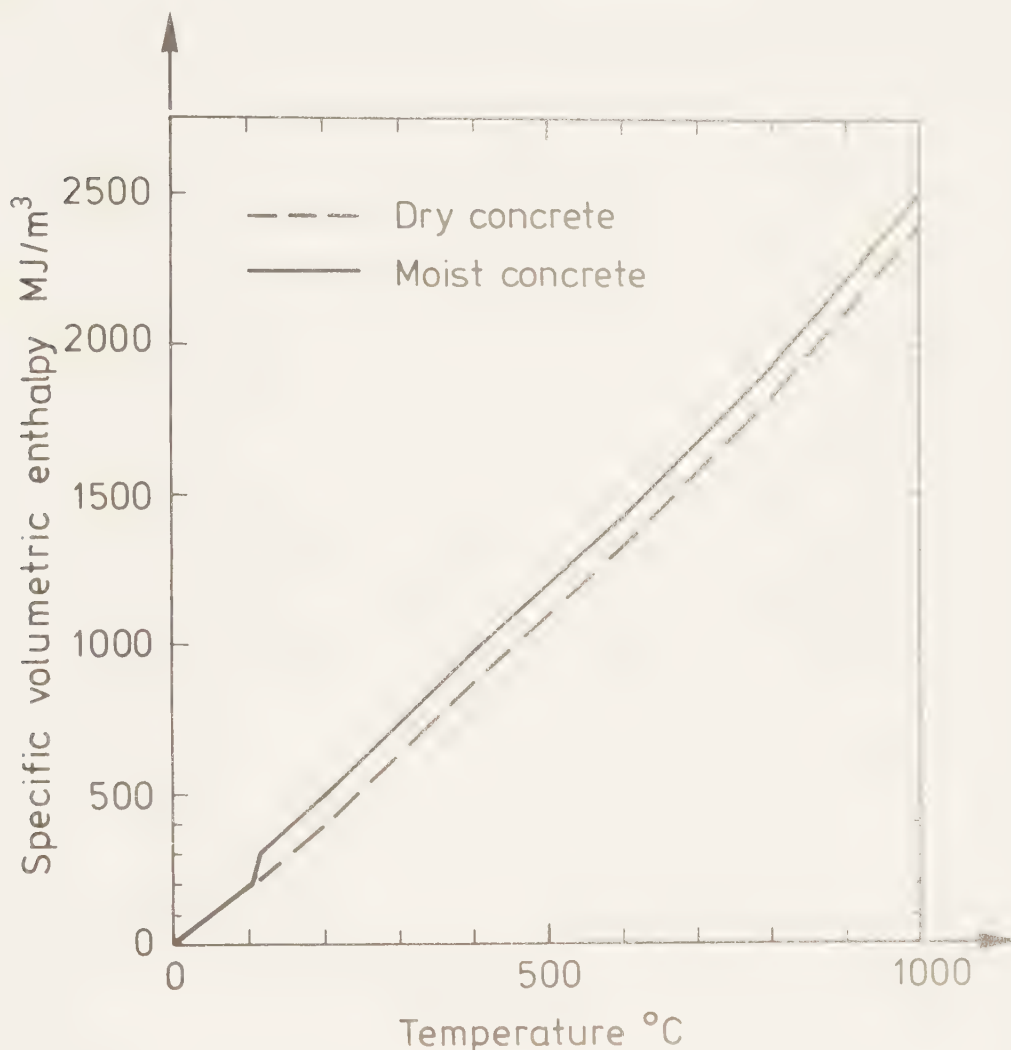


Figure 5.7. Specific volumetric enthalpy of dry concrete and concrete containing 1.5% water by weight

The cool side surfaces of the specimens were small in comparison with surrounding surfaces at ambient temperature. The resultant emissivity ϵ_r , as defined by Equation (2.15), was therefore chosen to be equal to the emissivity of appropriate material surfaces; i.e. 0.6 [17] and 0.8 [18]

for steel and concrete surfaces, respectively. On the fire-exposed side the same resultant emissivities were chosen, as radiation conditions in the furnace were to little known to justify any other values.

When assessing convection heat transfer factors β and powers γ as defined by Equation (2.14), free convection was assumed and the formula [19]

$$Nu = C (Pr Gr)^m \quad (a)$$

was employed. The Nusselt number Nu is defined by the equation

$$Nu = \frac{q^c \bar{d}}{k (T_s - T_g)} \quad (b)$$

where q^c is heat transferred by convection, $\bar{d}$ characteristic length, and k gas conductivity. The Prandtl number Pr is approximately 0.7 and the Grashof number Gr is

$$Gr = \frac{g \frac{1}{T_b} (T_s - T_g) \bar{d}^3}{\nu} \quad (c)$$

where g is the acceleration of gravity, $T_b = (T_s + T_g)/2$ the average absolute boundary layer temperature, and ν kinematic viscosity. For horizontal plates the characteristic length is calculated from

$$\bar{d} = \frac{A}{P}$$

where A is the area and P is the perimeter of the surface. Conductivity and viscosity of air are functions of temperature T_b [20]:

$$k = 13.75 \cdot 10^{-5} T_b^{0.92} \quad [W/mK] \quad (d)$$

$$\nu = 1.13 \cdot 10^{-9} T_b^{5/3} \quad [m^2/s] \quad (e)$$

From Equations (a-e) the formula (SI-units)

$$q^C = 13.75 \cdot 10^{-5} (5.48 \cdot 10^{18})^m C \frac{d^{3m-1}}{(\frac{13}{3}m - 0.92)} (T_s - T_g)^{m+1} \quad (f)$$

T_b

is derived.

The characteristic length d of the beams considered is approximately 0.15 m. Substitution of Equation (e) into Equation (c) then gives $Gr < 8 \cdot 10^6$ for expected levels of temperature. Laminar convection is therefore expected on the cool side, and C and m are 0.54 and 0.25, respectively [21]. Equation (f) then yields

$$q^C = 3.59 d^{-0.25} T_b^{-0.16} (T_s - T_g)^{1.25} \quad [W/m^2]$$

and by inserting $d = 0.15$ m and assuming $T_b = 400$ K, the convection factor β and power γ are identified as $2.2 W/m^2 K^{1.25}$ and 1.25, respectively. At the fire exposed side the burners will cause turbulent conditions [22], and therefore C and m are 0.15 and 1/3 [21], respectively. Thus

$$q^C = 36 T_b^{-0.52} (T_s - T_g)^{1.33} \quad [W/m^2]$$

and if T_b is assumed to be 1000K, β and γ are identified as $1.0 W/m^2 K^{1.33}$ and 1.33, respectively.

Convection heat transfer is only approximately modeled in TASEF-2. Errors in predicted temperature thus introduced are, however, negligible on the hot fire-exposed side, where radiation is dominant, while on the cool side they may be relatively great near the surface.

The beams were tested for one and a half hours; the furnace temperature approximately followed the ISO 834 standard time gas temperature curve for one hour and then it cooled off for half an hour.

Temperature in the furnace was measured with bare thermocouples. That is, however, not sufficient to determine accurately the heat transfer by radiation and convection from the furnace to the specimens [22]. The gas time-temperature curves assumed in the calculation were therefore adjusted so that calculated and measured temperature matched at the center of the fire-exposed flanges.

5.2.2 Example II - Wide-flange I-beam (Figure 5.2)

The finite element mesh shown in Figure 5.8 was employed to predict temperature in the wide-flange I-beam. Since the steel elements are small and the thermal diffusivity

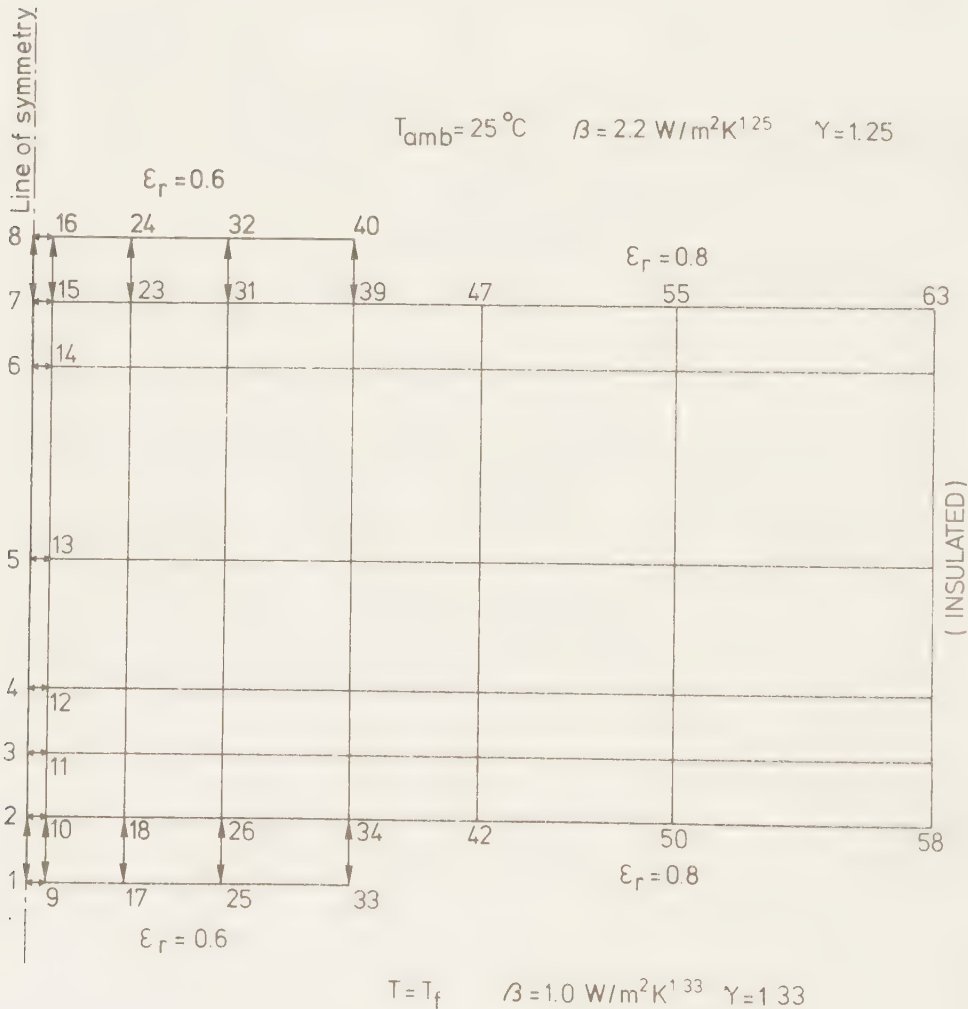


Figure 5.8. Finite element mesh of I-beam embedded in concrete

of steel is high, temperature differences between two nodes on opposite sides of a steel sheet will be negligible. Such nodes were therefore coupled as shown in Figure 5.8, and required to reach the same temperature. Critical time increments are thus substantially increased without introducing any noticeable error (see Section 3.4.2).

Assumed furnace gas temperature and measured and calculated temperature histories at the center of the top and bottom flanges, and along a vertical line 140 mm from the center-line of the steel cross section are plotted in Figures 5.9 and 5.10, respectively. Measured and calculated temperature distributions at selected times along the steel beam flanges and web are plotted in Figures 5.11 and 5.12, respectively.

While predicted steel temperature is accurate, such good agreement cannot be expected for concrete temperature since the effect of moisture migration is neglected. The characteristic plateau in the time-temperature curve at about 100°C , when water evaporates does, however, appear in the calculated temperature curve (Figure 5.10). Better agreement could be achieved if more accurate data on conductivity and specific enthalpy were available for the type of concrete tested.

5.2.3 Example III - Box girder (Figure 5.3)

The finite element mesh employed in the analysis of the box girder is shown in Figure 5.13. Only one half of the symmetrical cross section need be analyzed.

Heat transfer in the void by convection and radiation was considered as described in [10]. Convection heat transfer between the enclosure surfaces is assessed by neglecting heat capacity of enclosed air and assuming that no air

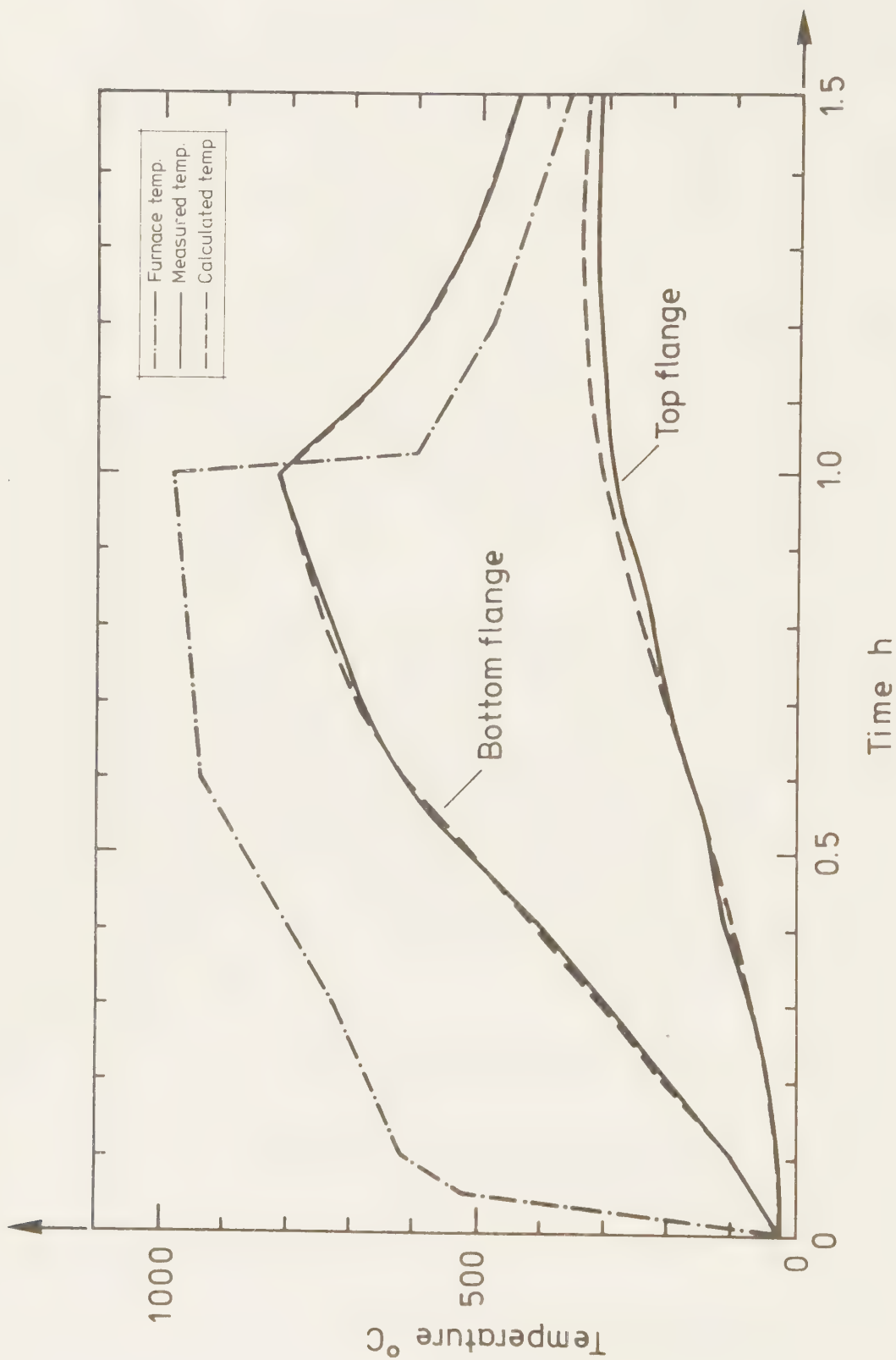


Figure 5.9. Assumed furnace gas temperature in Example II and measured and calculated temperature histories of top and bottom flanges at the centerline

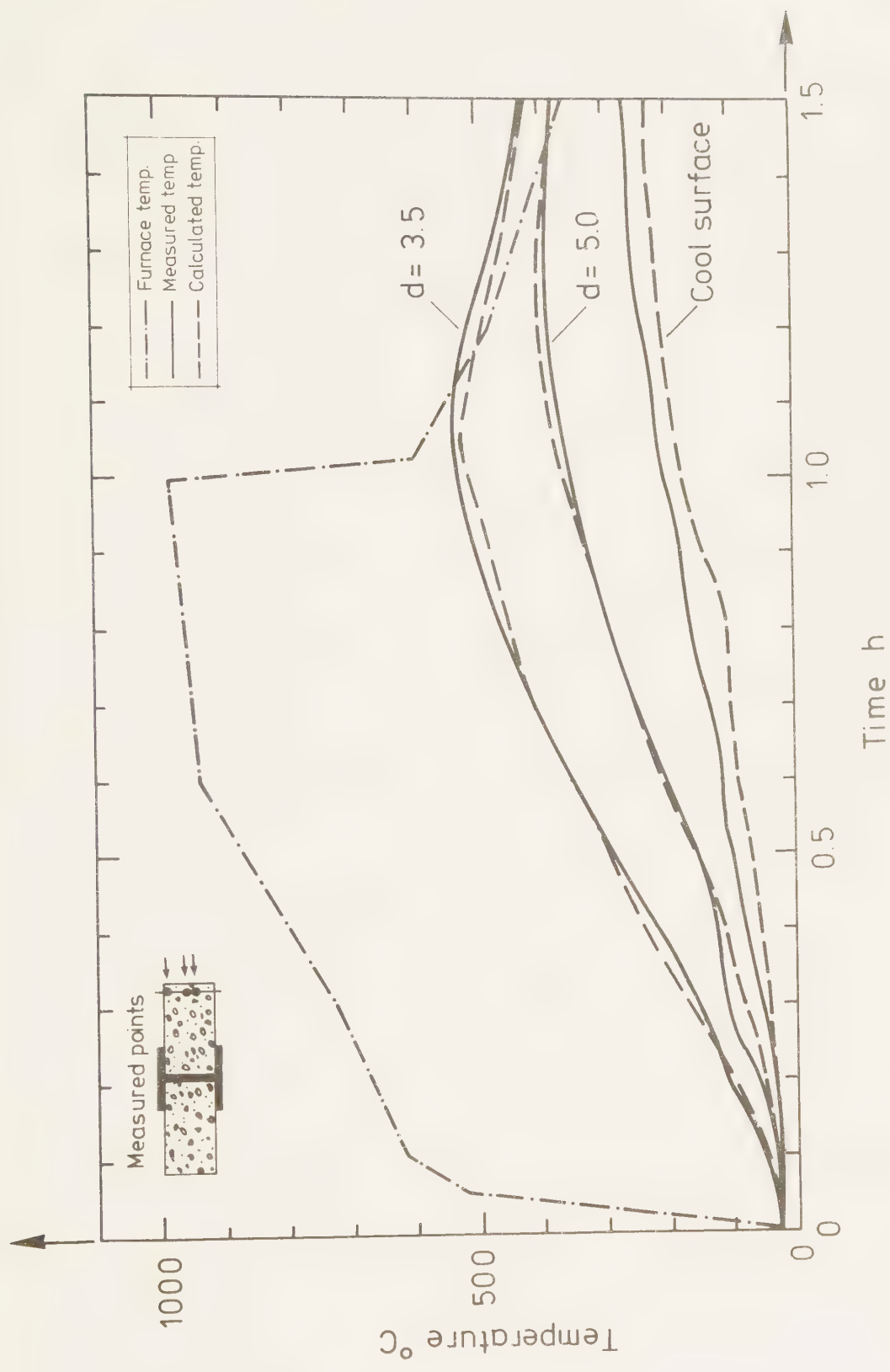


Figure 5.10. Calculated and measured concrete temperature at 140 mm from the centerline of steel beam and at distances d equal to 3.5 and 5.0 cm from fire-exposed surface and on cool surface, respectively

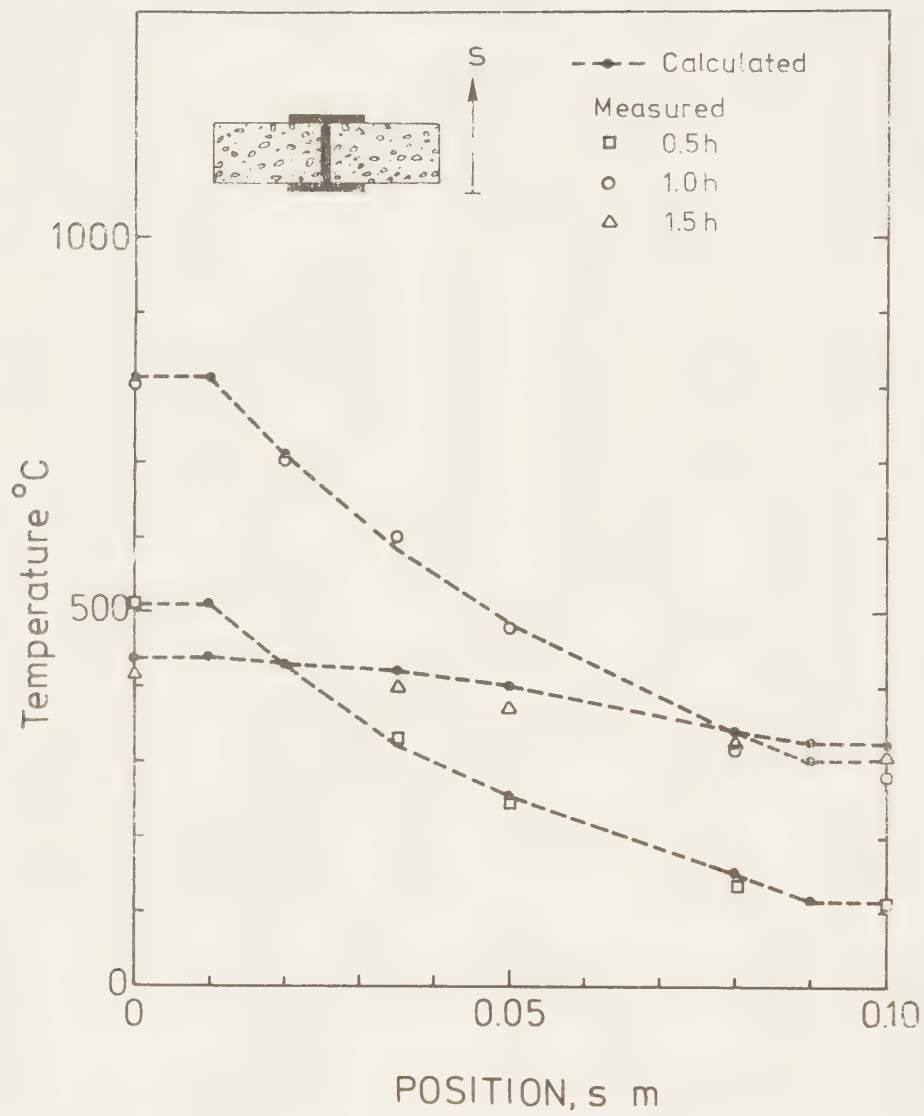


Figure 5.11. Calculated and measured temperature distributions along web of I-beam at selected time.

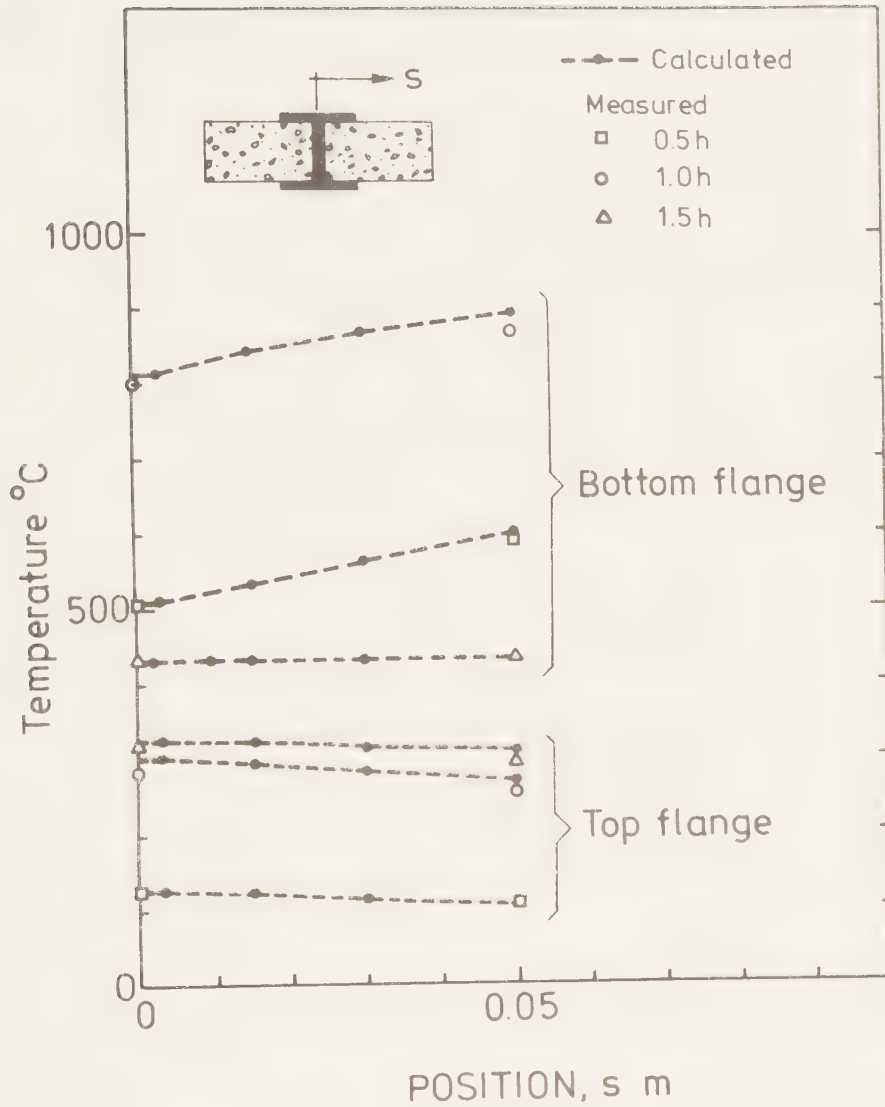


Figure 5.12. Calculated and measured temperature distributions along flanges of I-beam at selected times

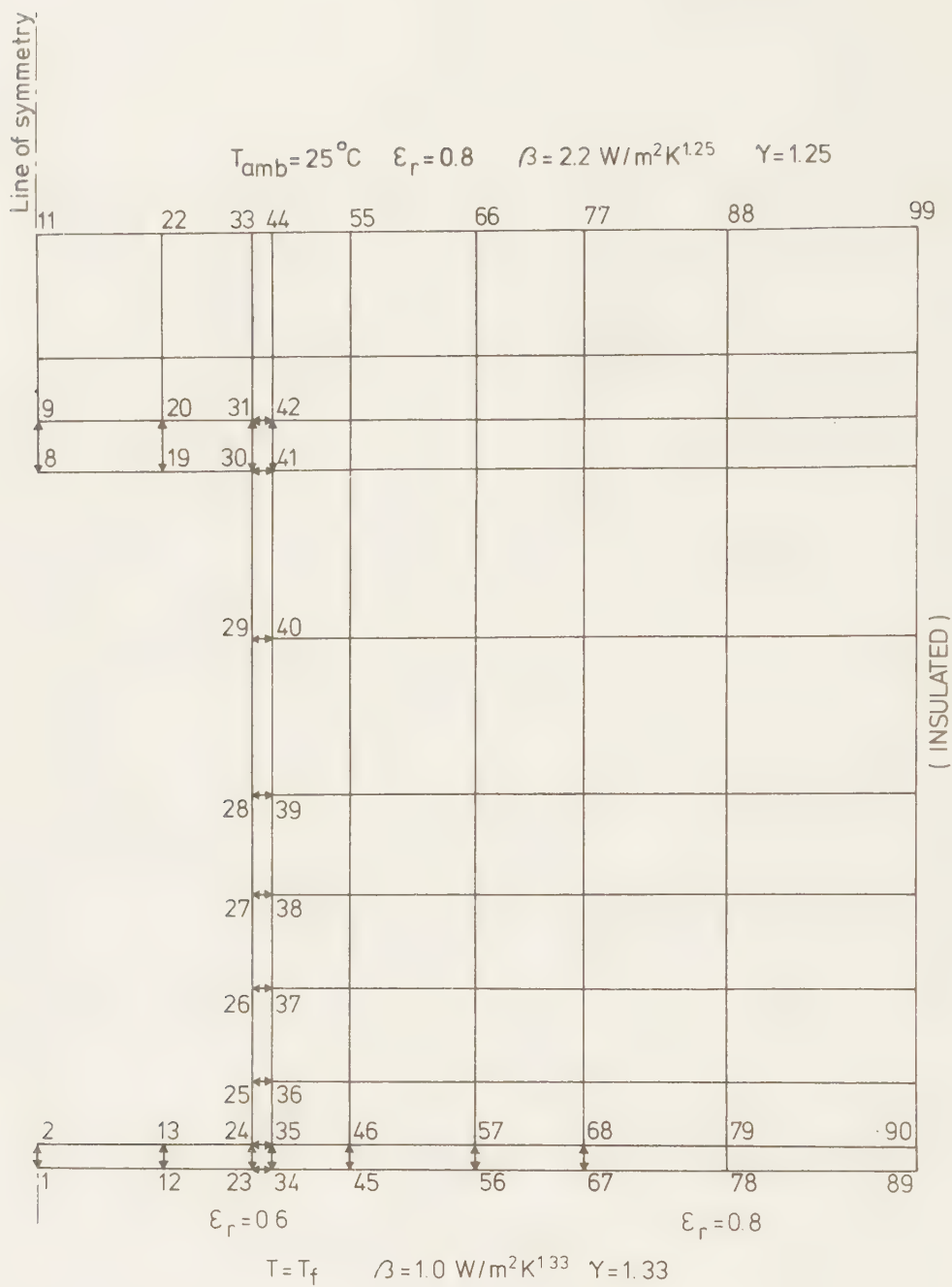


Figure 5.13. Finite element mesh of box girder embedded in concrete

flows along the beam. Convection factors β and powers γ for the enclosure surfaces are estimated by assuming free convection between two horizontal plates of uniform temperature. Thus β and γ equal to $1.6 \text{ W/m}^2 \text{ K}^{4/3}$ and 1.33, respectively, were obtained by assuming a temperature level of 500 K [19]. This estimation of the convective heat transfer is very rough, but any error introduced will be small as radiation increasingly dominates for increasing temperature. When calculating radiation heat transfer between the enclosure surfaces, temperature between adjacent nodal points is assumed to be uniform; by considering view factors and emissivities, the net radiation to each surface is calculated and then distributed to adjacent nodes [10]. The emissivity of the enclosure steel surfaces was assumed to be 0.6 [17].

In Figure 5.14 assumed furnace gas temperature and measured and predicted temperature of the center of the upper and lower flanges of the box girder and the cool upper concrete surface at the line of symmetry are plotted versus time. Distributions of measured and predicted temperature in the flanges and webs are plotted in Figures 5.15 and 5.16, respectively, at selected times.

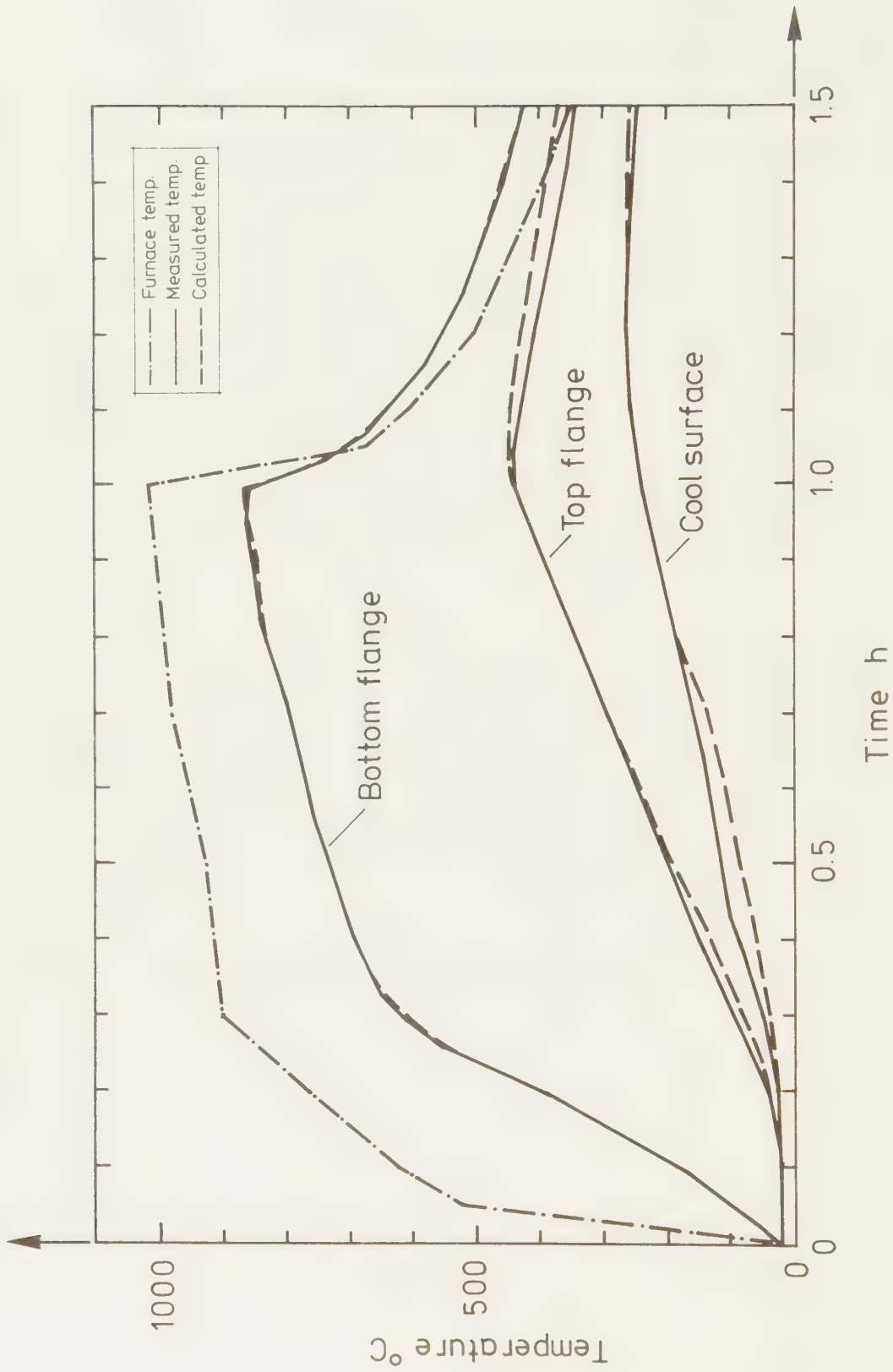


Figure 5.14. Assumed furnace gas temperature in Example III and measured and calculated temperature of top and bottom flanges of box girder and cool concrete surface at the centerline

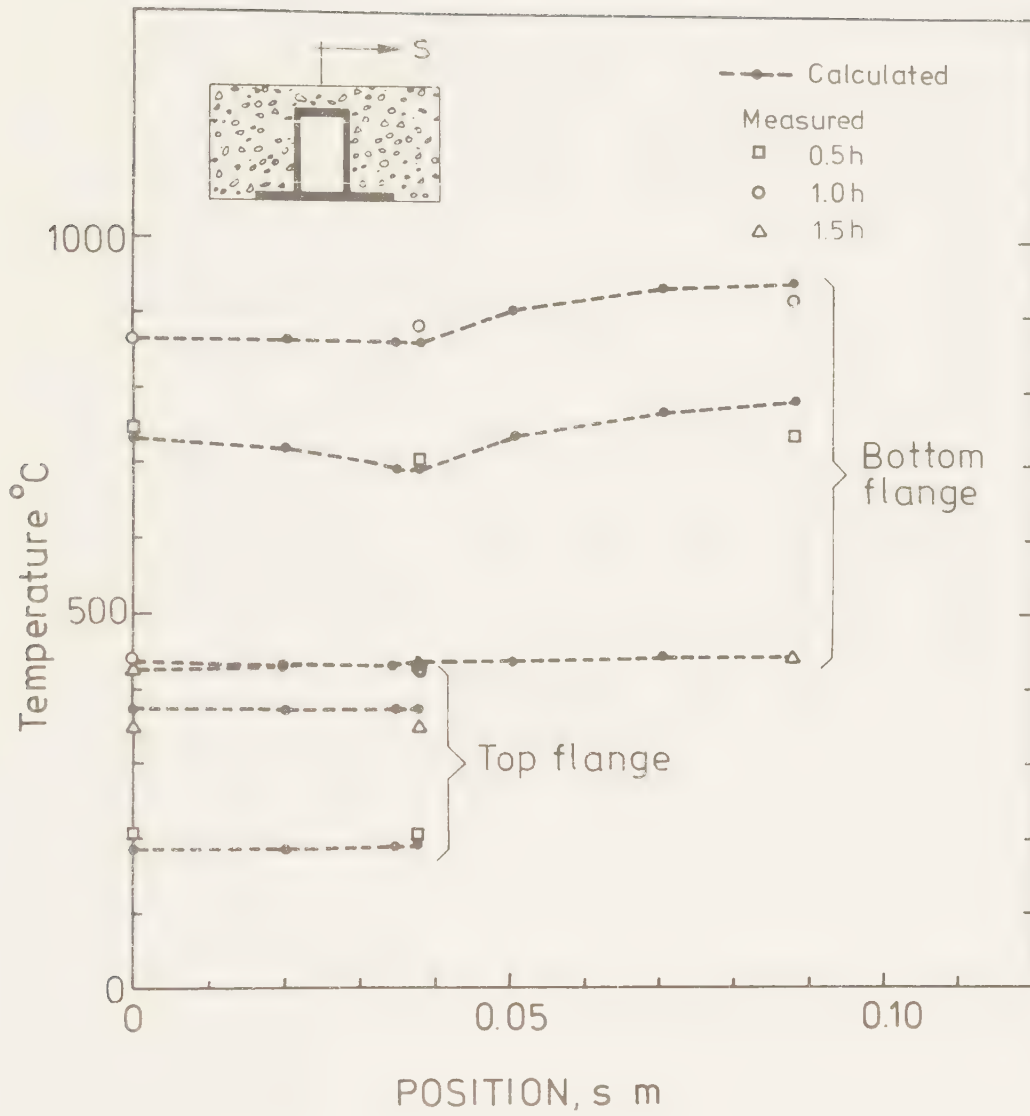


Figure 5.15. Calculated and measured temperature distributions along flanges of box girder at selected times

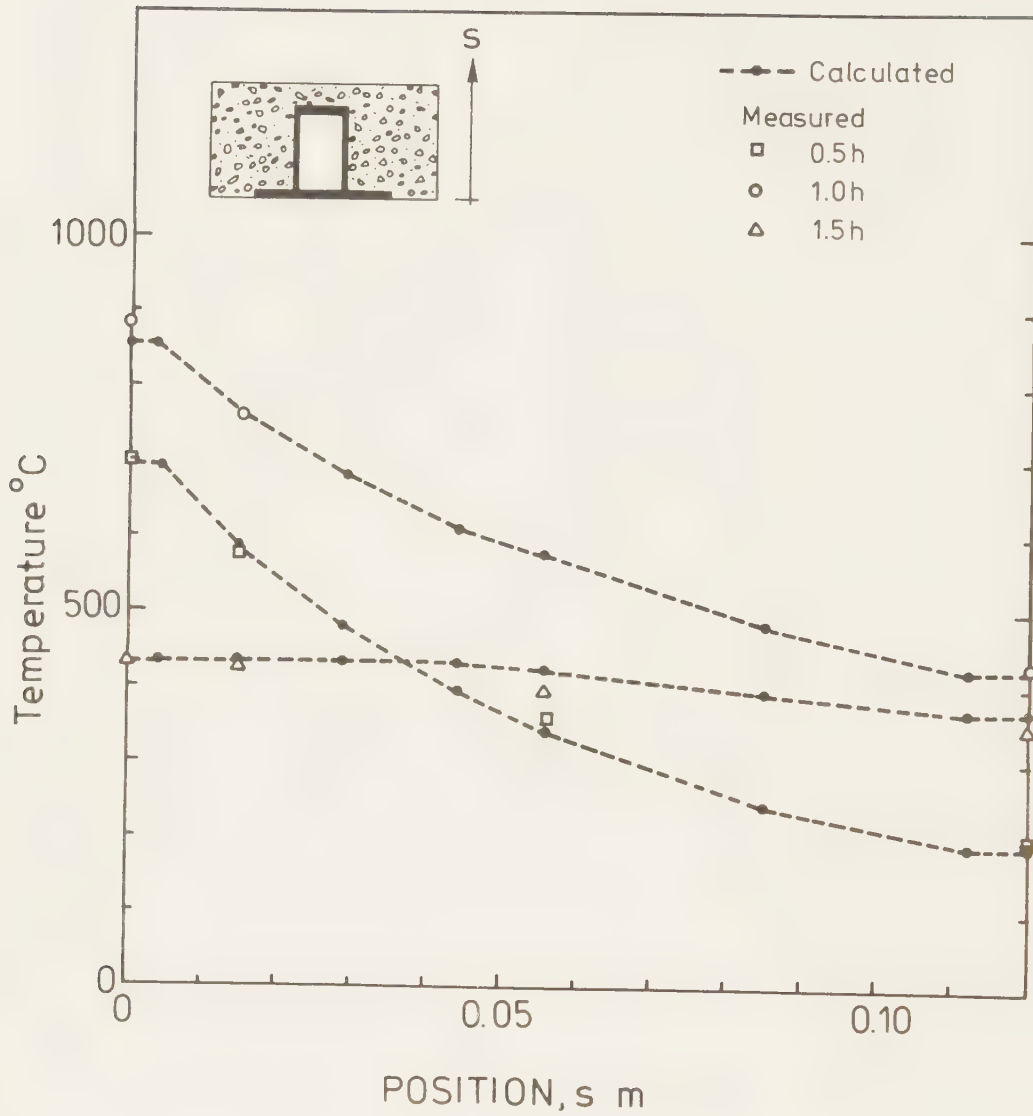


Figure 5.16. Calculated and measured temperature distributions along webs of box girder at selected times

6. SUMMARY AND CONCLUSIONS

6.1 Present Study

TASEF-2 is particularly well suited for the analysis of temperature in fire-exposed structures; the program is simple to use; rectangular finite element meshes are generated automatically with a minimum of input; nonlinearities due to the temperature dependence of material properties and boundary conditions can be considered; and heat transfer by radiation and convection in voids can be calculated using an algorithm described in [10].

The forward difference step-by-step time integration scheme used in TASEF-2 is a very efficient means of solving problems where materials having nonlinear temperature-specific heat relations - due to for instance evaporation of water - must be considered. A technique has been developed by which the critical time increment at which the applied step-by-step method will become numerically unstable can be estimated. Time increments are then calculated as a user-specified fraction of the critical time increment at each time step.

To avoid unnecessarily short time increments, and thus lengthly computations, nodes expected to attain approximately the same temperature are coupled and required to attain equal temperature. The technique has been successfully applied to composite cross sections of concrete and steel exposed to fire.

Three problems were analyzed in order to assess the accuracy and efficiency of TASEF-2. The solution of the first problem was compared to an analytical solution. The accuracy was good even for relatively coarse finite element meshes and long time increments. In the other two problems predicted temperature in composite steel and concrete beams was compared

to temperature measured during laboratory tests. One of the beams enclosed a void where heat transfer by radiation and convection was considered. The analysis proved to be accurate particularly for steel temperature. An equally good accuracy was not possible for concrete temperature because the thermal properties at elevated temperature of concrete are not as accurately known and the influence of mass transfer of water is not considered in the model. Heat of vaporization is, however, accounted for by stepping the temperature-specific enthalpy curve in the temperature range when water in the concrete evaporates (Figure 2.1). The total heat balance for a body heated to a temperature above the range at which evaporation occurs is therefore correct, and thus predicted temperature can be expected to be more accurate at high temperatures.

6.2 Future Development

In present version of TASEF-2, rectangular two-dimensional elements are available. Various one-, two-, and three-dimensional elements could be relatively easily introduced, but input would then be more complicated.

Heat transfer due to mass transfer in porous materials is not considered in TASEF-2. An extension of the model to include such phenomena would substantially complicate the analysis; in addition material data on mass diffusivity at high temperature are difficult to obtain. Results may, however, be improved with the present model if material properties determined at transient conditions accomplished by exposing specimens to boundary conditions similar to those in a fire were used in the analysis. A finite number of parameters by which the variation with temperature of one of the thermal properties - conductivity or specific enthalpy - could be described would then determine iteratively. Estimated parametric values are first used in such an analysis; calculated and measured temperature are then

compared. A new set of parameters is then chosen and new temperatures calculated. The procedure is repeated until the difference between measured and calculated temperature is minimized; computer programs are available by which the iterative search for parametric values can be accomplished. An optimal set of input data to the numerical model can thus be obtained for a given material, exposed to a similar fire; the influence of moisture migration will be indirectly considered in such an analysis.

The explicit forward difference time integration scheme used in TASEF-2 is very efficient for the nonlinear problem considered in this report. Nodes for which heat capacity is low and which are separated from adjacent nodes by little thermal resistance may, however, require that very small time increments be used if numerical stability is to be ensured. In the present version of TASEF-2, such critical nodes can be coupled and restrained to equal temperature, as illustrated in Example II and III in Section 5. If the error thus introduced is unacceptable, algorithms could be employed so that unconditionally stable implicit methods could be used for critical nodes. Such mixed implicit-explicit procedures have successfully been used in structural dynamics [23].

7. REFERENCES

- [1] Ödeen, K., "Fire Resistance of Prestressed Concrete Double T Units", Acta Polytechnica Scandinavica, Ci 48, Stockholm, 1968.
- [2] Wilson, E.L. and Nickell, R.E., "Applications of the Finite Element Method to Heat Conduction Analysis", Nuclear Engineering and Design, Vol. 4, 1966, pp. 276-286.
- [3] Wilson, E.L. and Farhoomand, I., "Non-Linear Heat Transfer Analysis of Axisymmetric Solids", Structural Engineering Laboratory, Report No. 71-6, University of California, Berkeley, April 1971.
- [4] Becker, J., Bizri, H. and Bresler, B., "FIRES-T - A Computer Program for the Fire Response of Structures - Thermal", Report No. UCB FRG 74-1, Fire Research Group, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1974.
- [5] Iding, R., Bresler, B. and Nizamuddin, Z., "FIRES-T3 - A Computer Program for the Fire Response of Structures - Thermal (Three Dimensional Version)", Report No. UCB FRG 77-15, Fire Research Group, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1977.
- [6] Rohsenow, W.M. and Hartnett, J.P., Handbook of Heat Transfer, McGraw-Hill, New York, 1973.
- [7] Pettersson, O., Magnusson, S.E. and Thor, J., "Fire Engineering Design of Steel Structures", Bulletin 52, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1976.

- [8] Zienkiewicz, O.C., The Finite Element Method in Structural and Continuum Mechanics, 3rd edition, McGraw-Hill, London, 1977.
- [9] Strang, G. and Fix, J.F., An Analysis of the Finite Element Method, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
- [10] Wickström, U., "A Numerical Procedure for Calculating Temperature in Hollow Structures Exposed to Fire", Report No. UCB FRG 77-9, Fire Research Group, Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, 1974. - see also "A Numerical Procedure for Calculating Temperature in Hollow Structures Exposed to Fire", Report No. 79:2, Department of Structural Mechanics, Lund Institute of Technology, Lund, 1979.
- [11] ISO, Fire-Resistance Tests - Elements of Building Construction, International Standard 834, 1975-11-01.
- [12] Carslaw, H.S. and Jaeger, J.C., Conduction of Heat in Solids, 2nd ed., Oxford University Press, 1969.
- [13] Alpsten, G., Egenspänningar i varmvalsade stålprofiler (Residual Stresses in Hot-rolled Steel Sections), Institutionen för Brobyggnad, Kungliga Tekniska Högskolan, Stockholm, 1967 (In Swedish).
- [14] Anderberg, Y., "Fire-Exposed Hyperstatic Concrete Structures - An Experimental and Theoretical Study", Bulletin 55, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1976.
- [15] Ödeen, K. and Nordström, Å., "Termiska egenskaper hos betong vid höga temperaturer (Thermal Properties of Concrete at Elevated Temperature)", Cement och Betong 1972:1, Stockholm, 1972 (in Swedish).

- [16] Ahlgren, L., "Fuktfixering i porösa byggnadsmaterial (Moisture Fixation in Porous Building Materials)", Report 36, Division of Building Technology, Lund Institute of Technology, Lund, 1972 (Swedish).
- [17] Schmidt, E. and Eckert, E., "Über die Richtungsverteilung der Wärmestrahlung von Oberflächen", Forsch.Ing.-Wes. 6, 1935.
- [18] Thor, J., "Strålningspåverkan på oisolerade eller undertaksisolerade stålkonstruktioner vid brand", Bulletin 29, Division of Structural Mechanics and Concrete Construction, Lund Institute of Technology, Lund, 1972 (in Swedish).
- [19] Holman, J.P. , Heat Transfer , 4th edition, McGraw-Hill, 1976.
- [20] Hadvig, S. , Stökiometri og Stofvaerdier (Stoichiometry and Material Properties) , Danmarks Tekniske Højskole, 1972 (in Danish).
- [21] Fujii, T. and Imura, H., "Natural Convection Heat Transfer from a Plate with Arbitrary Inclination", International Journal of Heat and Mass Transfer, Vol. 15, pp. 755, 1972.
- [22] Paulsen, O. , Om varmeovergang i Brandprøveovne (On Heat Transfer in Fire Test Furnaces) , Danmarks Tekniske Højskole, 1975, (in Danish).
- [23] Hughes, T.J.R. and Liu, W.K., "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory", Journal of Applied Mechanics, Vol. 45, June 1978.

APPENDIX A - USER'S MANUAL

The solution technique employed in TASEF-2 is summarized in Table A:1. Variable names are as in the program and differ occasionally from that used in previous sections. Table A:2 contains all subroutines in TASEF-2 with corresponding routine references, input variables, and common blocks; a chart of all routines is shown in Figure A:1. Detailed input instructions are given in Table A:3. Except for title cards, all input is read in free field format; input fields are then separated by a comma, or one or more blanks. Sequential commas are recognized as zero input values. Input variables may be given in any consistent unit system. Default values are, however, in SI-units and the expression for the ISO 834 standard test curve assumes time in hours and temperature in Kelvin or degree Celsius.

TABLE A:1

SUMMARY OF SOLUTION METHOD

(Variable names as in TASEF-2)

INITIAL CALCULATION:

1. Input geometry and dimension system arrays.
2. Define coupled nodes.
3. Input material data.
4. Input initial and ambient temperature.
5. Form node groups and input appropriate heat transfer data.
6. Define boundary node groups for prescribed heat transfer, and form heat transfer matrices BR and BC.
7. Define boundary node groups for prescribed temperature.
8. Define void node groups and form internal heat exchange matrices E and H.
9. Input time data.
10. Form node volume vector W.

FOR EACH FIRE TO BE ANALYZED:

11. Input new fire time temperature curve, or terminate program.
12. Initialize nodal temperature T and enthalpy EN vectors. Set the time variable TIME and the time step counter KTIME equal to zero.

TABLE A:1 (Cont.)

13. Set the time increment DELTI equal to zero and execute first time step for calculating first time increment only.

FOR EACH TIME STEP:

14. Increment time step: $TIME=TIME+DELTI$ and $KTIME=KTIME+1$.

15. Form new conduction matrix, and compute nodal heat flow F from nodes by internal conduction.

16. Find fire temperature.

17. Compute nodal heat flow by heat transfer at boundaries and in voids and internally generated.

18. a) For nodes surrounded by elements of one material, compute new nodal specific volumetric enthalpy
 $EN=EN+DELTI*(FLOW-F)/W$
and obtain new temperature from the temperature specific volumetric enthalpy relation of the material.

b) For nodes at interfaces between materials, compute new nodal enthalpy

$$EN=EN+DELTI*(FLOW-F)$$

and get new temperature by iteration

$$T^{i+1}=EN/P$$

where the total heat capacity of a node P is a function of temperature.

19. Set prescribed temperature to appropriate boundary nodes.

20. Print nodal temperature if required.

21. Test for more time steps:

YES: Calculate new time increment DELTI. Go to step 14.

NO : Print maximum temperature obtained during the process. Go to step 11.

TABLE A:2. Subroutines of TASEF-2

NR	ROUTINE	REFERENCED IN	REFERENCES	INPUT VARIABLES	COMMON BLOCKS
1	ACOUPL	FEM2			
2	AMB	PROG2			
3	ASSA2	FEM2	COND2	TINIT,TAMB,SIGMA,TABS	UNIT
4	ASSP2	FEM2	INTP,XVERSY		RGEO
5	ASSW2	PROG2			RMAT
6	BFIRE	INIT		TITFIR,NFP,TIM,TB	RGEO
7	BRBCA	FQBND A			UNIT,FIRE
8	BRBCB	FQBND B			
9	COND2	ASSA2	XVERSY		BNOD
10	COUPLA	PROG2		NCPLG;NCOUPL	RMAT
11	COUPLB	FEM2			COUPLE
12	COUPLC	FEM2			COUPLE
13	CTEMP	FEM2			COUPLE
14	DTIME	FEM2			TOUT
15	ENCLO1	PROG2	ENRAD1,ENCON1	CONTR0;NENC;XSYM, YSYM,IGREN	BNOD,ENCLOS,ENRAD, ENCON,DIM,UNIT ENCLOS
16	ENCLO2	FEM2	ENRAD2,ENCON2		BNOD,ENCLOS,ENCON, DUMMY
17	ENCON1	ENCLO1	HTRANS		ENCON,ENCLOS,DUMMY
18	ENCON2	ENCLO2			BNOD,ENCLOS,ENRAD, UNIT,DIM,DUMMY
19	ENRAD1	ENCLO1	VIEWFC,INVER,MULT, ETRANS		ENCLOS,ENRAD,UNIT, DUMMY
20	ENRAD2	ENCLO2	RADVEC		
21	ETRANS	ENRAD1	INIT,ASSP2,XVERSY, MINTP,ASSA2,MPACKV, FQBND B,FQGEN,ENCLO2, COUPLB,ACOUPL,CTEMP, HTEMP,PTBND B,COUPLC, OUT2,MAXCO,DTIME, OUTMA2		RMAT,FIRE
22	FEM2	PROG2	BRBCA BRBCB		
23	FQBND A	PROG2	NFQNG;FA1,ING1		UNIT,BNOD
24	FQBND B	FEM2			FQB,BNOD,UNIT

TABLE A:2. (Cont.)

NR	ROUTINE	REFERENCED IN	REFERENCES	INPUT VARIABLES	COMMON BLOCKS
25	FOGEN	FEM2			RMAT
26	GEOCO2	PROG2			ELFICT
27	HTEMP	FEM2	XVERSY		RMAT
28	HTRANS	ENCON1			
29	INIT	FEM2	BFIRE		TOUT,UNIT,FIRE
30	INTERF	PROG2			RGEO,COUPLE
31	INTP	ASSP2	XVERSY		RMAT
32	INVER	ENRAD1			
33	MAIN2				
34	MAT	PROG2	NET2,MESH2,PROG2		
				MAT;CCC,NTC,NTE,NQE, ET;TC,C;TE,ENT;TQ,QE	RMAT,RGEO
35	MAXCO	FEM2			TOUT
36	MINTP	FEM2			COUPLE
37	MESH2	MAIN2			
38	MPACKV	FEM2			
39	MULT	ENRAD1			DUMMY
40	NET2	MAIN2		TITLE;AXIAL,XMAX,YMAX, XBOX,YBOX,NR,NX,NY; ELFICT,SRDIAC;XA,YA NGROUP;NCHECK,NUMB, EPSG,BETA,GAMMA; NBOUND	RGEO
41	NGROUP	PROG2			BNOD,ENRAD,ENCON
42	OUT2	FEM2			TOUT
43	OUTMA2	FEM2			FIRE
44	PROG2	MAIN2	REG2,COUPLA,INTERF, MAT,GEOCO2,AMB,NGROUP, FQBND,PTBND,ENCLO1, TIME,TIME,ASSW2, COUPLB,FEM2		
45	PTBND	PROG2		NPTNG;FA1,ING1	PTB,BNOD
46	PTBNDB	FEM2			PTB,BNOD,UNIT
47	RADVEC	ENRAD2			
48	REG2	PROG2			RGEO
49	TIME	PROG2		NT,TIMMAX,DTMAX,TIMFAC, KTMAX,KUPDA;TOUT	
50	VIEWFC	ENRAD1			BNOD,ENCLOS
51	XVERSY	ASSP2,COND2, FEM2,HTEMP, INTP			

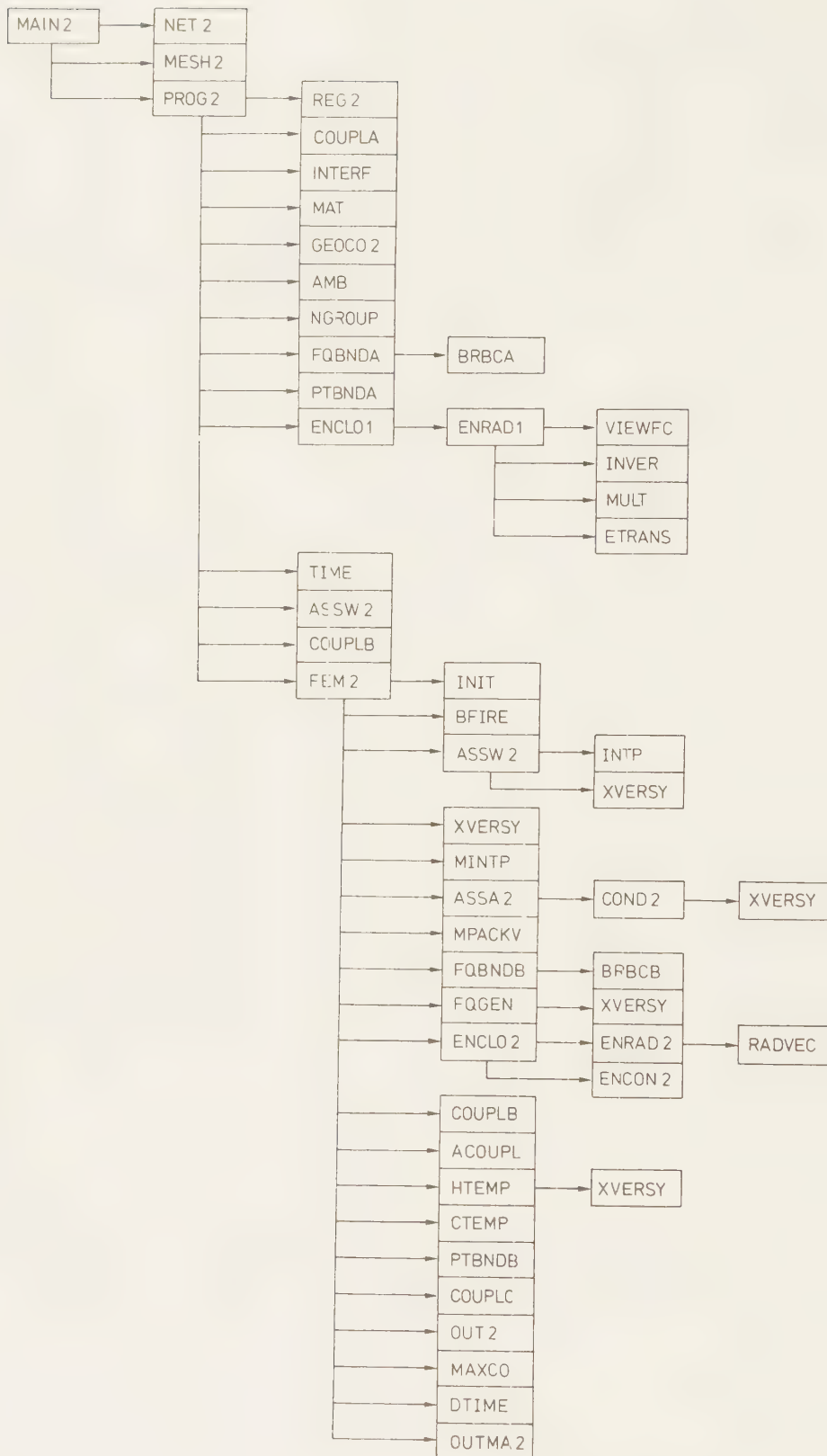


FIG A:1. Chart of subroutines in TASEF-2

TABLE A:3

Input Instructions

Input variables are given appropriate default values if zero is input.

A. TITLE CARD, FORMAT (20A4)

TITLE

Input appropriate title for labeling output.

B. GEOMETRY

1. Main geometry card

A base rectangle is generated between the coordinate axes and the lines $x = XMAX$ and $y = YMAX$. A number of subregions are defined by their minimum x- and y-coordinates, and maximum x- and y-coordinates. Lines parallel with the axes are generated at increments $XBOX$ and $YBOX$ or at reduced distances if subregions or specified lines to refine the mesh are present.

AXIAL, XMAX, YMAX, XBOX, YBOX, NR, NX, NY

AXIAL	.TRUE. or .FALSE. if axisymmetric or plane problem, respectively. (In present version of TASEF-2 axisymmetric problems cannot be analyzed)
XMAX	maximum x-coordinate of base structure
YMAX	maximum y-coordinate of base structure
XBOX	maximum distance between two x-lines (lines parallel to the y-axis)

TABLE A:3 (Cont.)

YBOX	maximum distance between two y-lines (lines parallel to the x-axis)
NR	number of regions ($NR \geq 1$). A structure is composed of a main region and a number of subregions of differing thickness or material properties. Fictitious subregions are used to specify voids and cut outs. Subregions will be defined by the following cards. (In current version $NR \leq 10$)
NX	number of specified x-lines for refining of element mesh
NY	number of specified y-lines for refining of element mesh

2. Subregion specifications

(NR-1) cards

ELFICT,SRDIAC(4)

ELFICT .TRUE. if the subregion is a void or cut out of the base structure. Otherwise .FALSE.

SRDIAC(4) minimum x- and y-coordinates, and maximum x- and y-coordinates of subregion

3. Specified x-lines

If $NX=0$ omit this card

XA(NX)

XA(NX) coordinates of specified x-lines

TABLE A:3 (Cont.)

4. Specified y-lines

If NY=0 omit this card

YA(NY)

YA(NY) coordinates of specified y-lines

As an example the structure in Figure A.2a is divided into a finite element mesh as shown in Figure A.2b by the following input cards: (The variable names are given within parentheses)

F,10.,6.,2.,1.5.,3,1,2 (AXIAL,XMAX,YMAX,XBOX,
 YBOX,NR,NX,NY)

F,3.,2.,6.,3.5 (ELFICT,SRDIAC(4))

T,6.,4.5,10.,6. (ELFICT,SRDIAC(4))

4.5 (XA(1))

1.0,5.25 (YA(1),YA(2))

C. COUPLED NODES

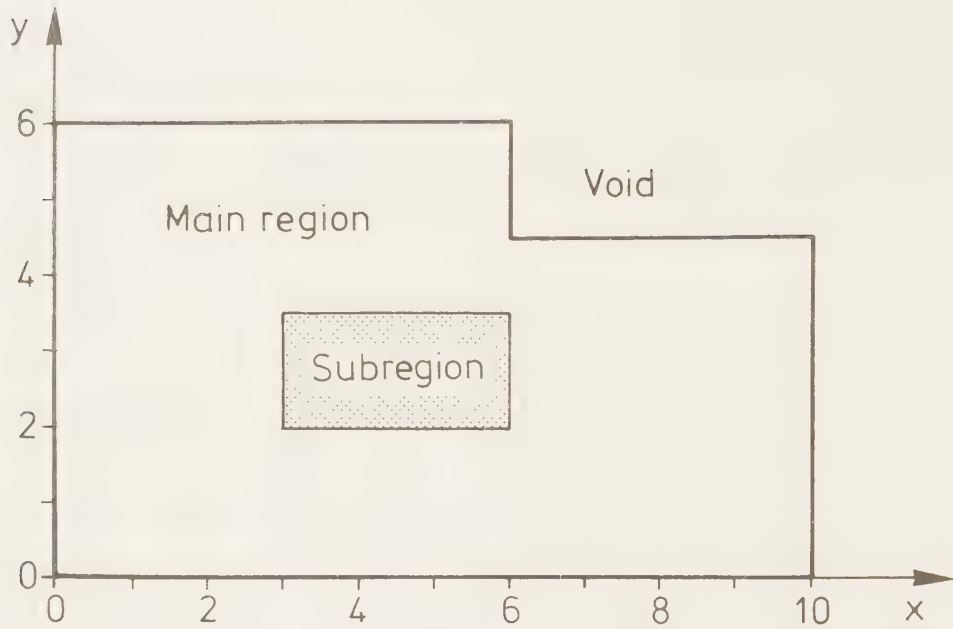
1. Number of groups of coupled nodes

NCPLG

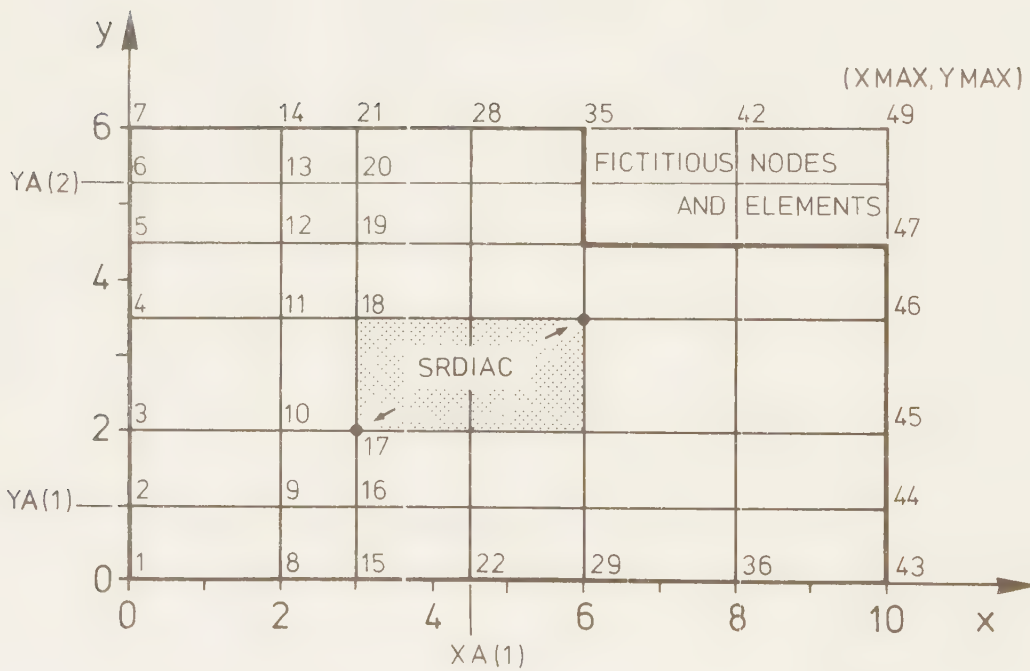
2. Each group

NCOUPL(8)

NCOUPL(8) coupled nodes. Each card must have 8 numbers. If fewer nodes in a group, fill with commas



(a) Structure to be analyzed



(b) Generated mesh with node numbers

Figure A.2. Example of mesh generation

D. MATERIAL DATA

For each nonfictitious region the following cards are read starting with the main region and followed by the subregion in the order as defined at B.2. Conductivity and enthalpy are input as sequential groups where each individual property is described as a piecewise linearized function of temperature.

1. Each material

a. Identification card

MAT

MAT arbitrary test to be written on output
list

b. Material description

CCC,NTC,NTE,NQE,ET

CCC .TRUE. if conductivity is constant in
cooling phase

NTC number of points associated with tempera-
ture conductivity function ($NTC \leq 20$)

NTE number of points associated with tempera-
ture specific volumetric enthalpy function
($NTC \leq 20$)

NQE number of points associated with tempera-
ture rate of heat generated per unit
volume function ($NQE \leq 20$)

ET thickness of elements; default 1

TABLE A:3 (Cont.)

c. Data card for temperature conductivity function

TC,C,TC,C,TC,C,... (NTC pairs)

The input is given in ordered pairs describing
each point (temperature, function value)

d. Data card for temperature specific volumetric
enthalpy function

TE,ENT,TE,ENT,TE,ENT,... (NTE pairs)

(same as c above)

e. Data card for time rate of heat generated per
unit volume function

(If NQE=0 omit this card)

TQ,QE,TQ,QE,TQ,QE,... (NQE pairs)

E. INITIAL AND AMBIENT TEMPERATURE, AND UNIT SYSTEM
DEPENDENT CONSTANTS

TINIT,TAMB,SIGMA,TABS

TINIT initial uniform temperature of structure

TAMB ambient temperature

SIGMA Stefan-Boltzmann constant; default
 SIGMA=5.67·10⁻⁸

TABS shift for absolute temperature; default
 273

TABLE A:3 (Cont.)

F. NODE GROUPS

Groups of nodes with common conditions are defined and, if appropriate, heat transfer properties at boundaries are specified.

1. Number of node groups

NGROUP

(NGROUP $\leq$ 10)

If NGROUP equal zero omit next card

2. Each node group

a. Properties

NCHECK, NUMB, EPSG, BETA, GAMMA

NCHECK node group number in sequential order
starting from 1

NUMB number of nodes of a group (NUMB $\leq$ 30)

EPSG emissivity

BETA convection factor

GAMMA convection power

b. Node numbers

NBOUND (NUMB)

TABLE A:3 (Cont.)

G. PRESCRIBED HEAT FLUX BOUNDARY

Boundary conditions are defined by node groups and their heat transfer properties. Surrounding gas is either at fire or ambient temperature.

1. Number of boundary node groups with prescribed heat flux

NFQNG

If NFQNG equals zero omit next card

2. Each boundary node group

FA1,ING1

FA1	.TRUE. if specified boundary temperature varies with time, e.g. fire temperature history .FALSE. if specified boundary temperature is the constant ambient temperature TAMB
-----	--

ING1	node group number
------	-------------------

H. PRESCRIBED TEMPERATURE BOUNDARY

Node groups with prescribed temperature are input.

1. Number of boundary node groups with prescribed temperature

NPTNG

If NPTNG equals zero omit next card

TABLE A:3 (Cont.)

2. Each boundary node group

FA1,ING1

Same as G.2

I. VOIDS

One or two voids with heat exchange between the enclosure surfaces may be defined by surrounding node groups. Heat exchange between enclosure surfaces is described by properties assigned to the node groups. Halves or quarters of voids may be analyzed if they are symmetrical around one or both of the coordinate axes.

1. Control card

CONTRO

If no voids exist insert arbitrary card, and omit the following cards. Otherwise input VOID.

2. Number of voids

NENC

(NENC $\leq$ 2)

3. For each void

a. XSYN,YSYM,IGREN(4)

XSYM .TRUE. if void is symmetric around the
 x-axis

TABLE A:3 (Cont.)

YSYM	.TRUE. if void is symmetric around the y-axis
IGREN(4)	node groups surrounding a void. If less than 4, fill with commas

J. TIME

1. Time control card

NT, TIMMAX, DTMAX, TIMFAC, KTMAX, KUPDA

NT	number of specified times for printing out of temperature
TIMMAX	maximum time of analysis
DTMAX	maximum time increment; default TIMMAX
TIMFAC	time increment factor; default 0.8
KTMAX	maximum number of time steps; default 1000
KUPDA	number of time steps between updating of heat conduction matrix; default 1

2. Specified times for temperature output

TOUT(NT)

K. FIRE TEMPERATURE HISTORY

1. Control card

TITFIR

TABLE A:3 (Cont.)

TITFIR is printed for identification of assumed fire. If TITFIR = 'ISO 834' the time temperature relation according to ISO 834 fire resistance standard test is assumed, and the next two cards are omitted. Terminate analysis by a blank card.

2. Fire temperature

A fire temperature history is input by a number of points on the time temperature curve. Temperature between these points are obtained by linear interpolation.

a. Number of points

NFP

(NFP $\leq$ 50)

b. Data card

TIM,TB,TIM,TB,TIM,TB,... (NFP pairs)

3. Go back to K1 and begin analysis with new fire, or terminate program by inserting blank card.

APPENDIX B - Listing of TASEF-2

The subroutines are listed in alphabetic order. The main program MAIN2 is coded in ALGOL and all other routines in FORTRAN V. Definitions of major variables are given in subroutine PROG2. Although the program has been tested, no warranty is made regarding its accuracy or reliability, and no responsibility is assumed in this respect.

```

1      SUBROUTINE ACOUPL(A,DTA,NN,MAX)
2      C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
3      C-----FOR EACH COUPLING NODE GROUP
4      DIMENSION A(NN,MAX),DTA(NN)
5      PARAMETER NCP=50
6      COMMON /COUPLE/ NCOUPL(NCP,8),NCPLG
7      C-----
8      IF (NCPLG.EQ.0) RETURN
9      DO 50 I=1,NCPLG
10     MNOD=NCOUPL(I,1)
11     AMX=0.
12     DO 40 J=1,8
13     NODJ=NCOUPL(I,J)
14     IF (NODJ.EQ.0) GOTO 50
15     DO 30 K=1,8
16     NODK=NCOUPL(I,K)
17     IF (NODK.EQ.0) GOTO 40
18     NDUM=NODJ-NODK
19     IF (NDUM) 30,30,60
20     60 CONTINUE
21     IF (NDUM.GE.MAX) GOTO 30
22     AMX=AMX+2.*A(NODJ,MAX-NDUM)
23     30 CONTINUE
24     C-----
25     40 AMX=AMX+A(NODJ,MAX)
26     50 DTA(MNOD)=DTA(MNOD)+AMX-A(MNOD,MAX)
27     RETURN
28     END
29
30      SUBROUTINE AMR
31      C-----READ INITIAL AND AMBIENT TEMPERATURE,
32      C-----AND UNIT DEPENDENT CONSTANTS
33      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
34      DATA SIGM,TAB/5.77E-8,273./
35      PRINT 200
36      C-----
37      READ 100,TINIT,TAMB,SIGMA,TABS
38      C-----
39      IF (SIGMA.LT.1.E-20) SIGMA=SIGM
40      IF (TABS.LT.1.E-20) TABS=TAB
41      PRINT 210,TINIT,TAMB,SIGMA,TABS
42      TAMB4=(TAMB+TABS)**4
43      RETURN
44      100 FORMAT(
45      200 FORMAT(/' INITIAL DATA'/1X,12(1H*))
46      210 FORMAT(/' INITIAL TEMPERATURE=' ,G9.3/
47      1 ' AMBIENT TEMPERATURE=' ,G9.3/
48      2 ' STEFAN-BOLTZMANN CONSTANT=' ,G9.3/
49      3 ' ABSOLUTE TEMPERATURE SHIFT=' ,G9.3)
50      END
51
52      SUBROUTINE ASSA2(NN,NE,N,KTOP,X,Y,FLA,T,TT,TMAX,AXIAL,MAX,A)
53      C-----THIS SUBROUTINE COMPUTES THE GLOBAL HEAT CONDUCTION MATRIX A
54      C-----
55      C----- NN NUMBER OF NODES
56      C----- NE NUMBER OF ELEMENTS
57      C----- N ELEMENT REGION NUMBER
58      C----- ELA GEOMETRIC DUMMY VECTOR
59      C----- T TEMPERATURE VECTOR
60      C----- TT TEMPERATURE HISTORY VECTOR
61      C----- TMAX .TRUE. IF MAXIMUM TEMPERATURE IS OBTAINED
62      C----- A GLOBAL HEAT CONDUCTION MATRIX
63      PARAMETER MNP=10
64      COMMON/RGEO/FLFICT(MNR),ET(MNR),SRDIAC(4,MNR)

```



```

54      DIMENSION N(NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),KTOP(4,NE),
55      1      A(NN,MAX),FLA(4,NE)
56      LOGICAL TMAX,ELFICT,AXIAL
57      DO 5 I=1,NN
58      DO 5 J=1,MAX
59      5      A(I,J)=0.
60      DO 10 I=1,NE
61      N1=N(I)
62      IF(ELFICT(N1)) GOTO 10
63      K1=KTOP(1,I)
64      K2=KTOP(2,I)
65      K3=KTOP(3,I)
66      K4=KTOP(4,I)
67      CALL COND2(T(K1),T(K2),T(K3),T(K4),TT(K1),TT(K2),TT(K3),TT(K4),
68      1      TMAX(K1),TMAX(K2),TMAX(K3),TMAX(K4),N(I),AXIAL,C)
69      A(K1,MAX)=A(K1,MAX)+C*ELA(1,I)
70      A(K2,MAX)=A(K2,MAX)+C*ELA(1,I)
71      A(K3,MAX)=A(K3,MAX)+C*ELA(1,I)
72      A(K4,MAX)=A(K4,MAX)+C*ELA(1,I)
73      A(K2,MAX-1)=A(K2,MAX-1)+C*ELA(2,I)
74      A(K3,MAX-K3+K1)=A(K3,MAX-K3+K1)+C*ELA(3,I)
75      A(K3,MAX-K3+K2)=A(K3,MAX-K3+K2)+C*ELA(4,I)
76      A(K4,MAX-K4+K1)=A(K4,MAX-K4+K1)+C*ELA(4,I)
77      A(K4,MAX-K4+K2)=A(K4,MAX-K4+K2)+C*ELA(3,I)
78      A(K4,MAX-1)=A(K4,MAX-1)+C*ELA(2,I)
79      10      CONTINUE
80      RETURN
81      END
82
83      SUBROUTINE ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,
84      1      NODINT,AXIAL)
85      C-----FORM HEAT CAPACITY MATRIX
86      DIMENSION N(1),X(1),T(1),TT(1),TMAX(1),EV4(4,1),NODEL(4,NN),
87      1      MNODEL(1),P(1),W(1),NODINT(1)
88      PARAMETER MNV=20,MNR=10
89      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
90      1      ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
91      LOGICAL CCC,LQ
92      LOGICAL AXIAL,TMAX
93
94      C
95      DO 30 I=1,NN
96      NODINI=NODINT(I)
97      IF(NODINI.LT.0) GOTO 30
98      TI=T(I)
99      IF(NODINI.GT.0) GOTO 20
100      P(I)=0.
101      CALL INTP(MNODEL(I),NODEL(1,I),V,EV4,TI,P(I))
102      GOTO 30
103      20      CONTINUE
104      CALL XVERSY(TE,ENT,MNV,NODINI,TI,ENI)
105      CRA=ENI/TI
106      P(I)=W(I)*CRA
107      30      CONTINUE
108      RETURN
109      END
110
111      SUBROUTINE ASSW2(NN,NE,N,KTOP,X,Y,EV4,AXIAL,W)
112      C-----THIS SUBROUTINE COMPUTES THE GLOBAL VOLUME VECTOR W
113      DIMENSION X(NN),Y(NN),KTOP(4,NE),EV4(NN),W(NN),N(NE)
114      PARAMETER MNR=10
115      COMMON/RGEO/ELFICT(MNR)
116      LOGICAL AXIAL,ELFICT
117      DO 5 I=1,NN
118      5      W(I)=0.

```

```

137         GO TO 147,14
138         M1=N(I)
139         IF (CLIFF(I,M1), CLIFF=1)
140         M1=KTOP(I,1)
141         K2=KTOP(I,1)
142         K=KTOP(I,1)
143         K4=KTOP(4,I)
144         V(X1)=X(X1)+EV4(I)
145         W(K2)=W(K2)+EV4(I)
146         X(X1)=X(X1)+EV4(I)
147         V(X4)=V(X4)+EV4(I)
148         CONTINUE
149         RETURN
150
151 SUBROUTINE BFIPE(FIN)
152 C-----THIS ROUTINE FORMS VECTORS FOR FIRE BOUNDARY
153 C-----TIME TEMPERATURE RELATION
154 COMMON/COMMON/TAHS,TINIT,TAMH,TAMH4
155 COMMON/FIRE/TIM(SO),TS(SO),TITFIR
156 LOGICAL FIN
157 INTEGER TITFIR(15),BLANK(15),ISO(2)
158 DATA BLANK/10*4H /,ISO/4HISO,4H534 /
159 FIN=.FALSE.
160 C-----INPUT TITLE OF FIRE BOUNDARY
161 C-----IF = BLANK = TERMINATE RUN
162 C-----IF = ISO 534 = STANDARD FIRE ASSUMED
163 C-----ELSE INPUT TIME TEMPERATURE PAIRS
164 READ 110,TITFIR
165 DO 10 I=1,15
166 IF (TITFIR(I).NE.BLANK(I)) GOTO 40
167 FIN=.TRUE.
168 RETURN
169 CONTINUE
170 PRINT 200
171 PRINT 210,TITFIR
172 DO 30 I=1,7
173 IF (TITFIR(I).NE.ISO(I)) GOTO 50
174 C-----ISO STANDARD CURVE
175 T1=0.
176 DO 40 I=1,20
177 TIM(I)=T1
178 TB(I)=T1+1000.-400.*EXP(-.2*T1)-270.*EXP(-1.7*T1)-
179 1 400.*EXP(-19.*T1)
180 40 T1=TI+DT
181 T1=TI-DT
182 TI=0.7 * T1
183 T1=TI+TI
184 DO 45 I=21,50
185 TIM(I)=TI
186 TB(I)=TIM(I)+1000.-400.*EXP(-.2*TI)-270.*EXP(-1.7*TI)-
187 1 400.*EXP(-19.*TI)
188 45 TI=TI+DT
189 RETURN
190 CONTINUE
191 C-----ARBITRARY FIRES
192 C-----INPUT NUMBER OF TIME TEMPERATURE PAIRS
193 READ 100,NFP
194 IF (NFP.LG.0) FIN=.TRUE.
195 IF (FIN) RETURN
196 C-----INPUT TIME TEMPERATURE PAIRS
197 READ 100,(TIM(I),TB(I),I=1,NFP)
198 PRINT 220,(TIM(I),TB(I),I=1,NFP)

```

```

192 100 FORMAT(
193 110 FORMAT(2044)
194 200 FORMAT(// ' FIRE BOUNDARY TEMPERATURE'//1X,25(1H*))
195 210 FORMAT(//1X,2044)
196 220 FORMAT(// ' FIRE BOUNDARY TIME - TEMPERATURE INPUT PAIRS'//
197 1 ' TIME',5X,' TEMPERATURE'// (510.3,510.3))
198 RETURN
199 END
200
201 SUBROUTINE BRBCA(BR,BC,EPSIG,BET,BAR,NUM1,NB,ING1)
202 C-----FORM BOUNDARY RADIATION AND CONVECTION MATRICES
203 DIMENSION BR(NUM1,2),BC(NUM1,2),BAR(NB,NUM1)
204 BR(1,1)=0.
205 BR(1,2)=.3333333*BAR(ING1,2)
206 NUM1=NUM1-1
207 IF(NUM1.EQ.1) GOTO 2C
208 DO 10 I=2,NUM1
209 BR(I,1)=.1666667*BAR(ING1,I)
210 BR(I,2)=.3333333*(BAR(ING1,I)+BAR(ING1,I+1))
211 10 CONTINUE
212 2C CONTINUE
213 BR(NUM1,1)=.1666667*BAR(ING1,NUM1)
214 BR(NUM1,2)=.3333333*BAR(ING1,NUM1)
215 C
216 DO 30 I=1,NUM1
217 DO 30 J=1,2
218 BC(I,J)=BET*BR(I,J)
219 BR(I,J)=EPSIG*BR(I,J)
220 30 CONTINUE
221 RETURN
222 END
223
224 SUBROUTINE BRPCB(BR,BC,TR,TC,TRD,TCD,NUM1,DTA,NN,MAX,FLOW,TG,
225 1 T,I,31)
226 C-----THIS ROUTINE CALCULATES EXTERNAL HEAT FLOW BY RADIATION AND CONVECTION
227 C-----AND ADDS THE CORRESPONDING CONTRIBUTIONS TO THE VECTOR DTA FOR
228 C-----CALCULATION OF CRITICAL TIME INCREMENT
229 DIMENSION BR(NUM1,2),BC(NUM1,2),DTA(4N),FLOW(NN),T(NN)
230 1 TR(NUM1),TC(NUM1),TRD(NUM1),TCD(NUM1)
231 PARAMETER NB=10,NNP=30,NNB2=2*NNB
232 COMMON/BNDB/NUMB(NB),NBOUND(NB,4NB),BAREA(NB,NNB),
233 1 EPSIG(NB),BETA(NB),CPG(NB),FA(NB)
234 LOGICAL FA
235 C
236 C-----FIRST NODE
237 C
238 NODE=NBOUND(ING1,1)
239 TR2=TR(1)
240 TC2=TC(1)
241 TR3=TR(2)
242 TC3=TC(2)
243 TRD2=TRD(1)
244 TCD2=TCD(1)
245 TRD3=TRD(2)
246 TCD3=TCD(2)
247 BR2=BR(1,2)
248 BC2=BC(1,2)
249 BR3=BR(2,1)
250 BC3=BC(2,1)
251 FLW=BR2*TR2+BC2*TC2
252 FLW=FLW+BR3*TR3
253 FLW=FLW+BC3*TC3
254 DA=BR2*TRD2+BC2*TCD2
255 DA=DA+BR3*TRD3

```

```

253      DA=DA+BC3*TC3
257      DTA(NODE)=DTA(NODE)+.5*DA
258      FLOW(NODE)=FLOW(NODE)+FLW
259
260      C-----INTERMEDIATE NODES
261      C
262      NUM1=NUMI-1
263      IF(NUM1.EQ.1)GOTO 2C
264      DO 1C I=2,NUM1
265      NODE=NBOUND(ING1,I)
266      TP1=TR2
267      TR2=TR3
268      TP3=TR(I+1)
269      TC1=TC2
270      TC2=TC3
271      TC3=TC(I+1)
272      TRD1=TRD2
273      TRD2=TRD3
274      TRD3=TRD(I+1)
275      TCD1=TCD2
276      TCD2=TCD3
277      TCD3=TCD(I+1)
278      BR1=BR3
279      BR2=BR(I,2)
280      BR3=BR(I+1,1)
281      BC1=BC3
282      BC2=BC(I,2)
283      BC3=BC(I+1,1)
284      FLW=BR2*TR2+BC2*TC2
285      FLW=FLW+BR1*TR1+BC3*TR3
286      FLW=FLW+BC1*TC1+BC3*TC3
287      DA=BR2*TRD2+BC2*TCD2
288      DA=DA+BR1*TRD1+BR3*TRD3
289      DA=DA+BC1*TCD1+BC3*TCD3
290      DTA(NODE)=DTA(NODE)+.5*DA
291      FLOW(NODE)=FLOW(NODE)+FLW
292
293      1C CONTINUE
294      C
295      C-----LAST NODE
296      C
297      NODE=NBOUND(ING1,NUMI)
298      TR1=TR2
299      TR2=TR3
300      TC1=TC2
301      TC2=TC3
302      TRD1=TRD2
303      TRD2=TRD3
304      TCD1=TCD2
305      TCD2=TCD3
306      BR1=BR3
307      BR2=BR(NUMI,2)
308      BC1=BC3
309      BC2=BC(NUMI,2)
310      FLW=BR2*TR2+BC2*TC2
311      FLW=FLW+BR1*TR1+BC1*TC1
312      DA=BR2*TRD2+BC2*TCD2
313      DA=DA+BR1*TRD1+BC1*TCD1
314      DTA(NODE)=DTA(NODE)+.5*DA
315      FLOW(NODE)=FLOW(NODE)+FLW
316      RETURN
317      END
318
319      SUBROUTINE COND(T1,T2,T3,T4,TT1,TT2,TT3,TT4,TMAX1,TMAX2,TMAX3,
1          TMAX4,N1,AXIAL,CE)

```

```

320 C-----GET ELEMENT CONDUCTIVITY
321 LOGICAL TMAX1,TMAX2,TMAX3,TMAX4,AXIAL
322 PARAMETER MNV=20,MNR=10
323 COMMON/RYAT/(CCC(MNR),T((MNV,MNF),C(MNV,MNR),TE(MNV,MNR),
324 1 ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LG(MNR)
325 LOGICAL CCC,LQ
326 IF(CCC(N1)) TM=(TT1+TT2+TT3+TT4)/4.
327 IF(.NOT.CCC(N1)) TM=(T1+T2+T3+T4)/4.
328 CALL XVERSY(TC,C,MNV,N1,TM,CE)
329 RETURN
330 END
331
332 SUBROUTINE COUPLA(NODCPL,NN,NODINT)
333 C-----READ COUPLED NODES AND FORM CONTROL VECTOR VCOUPL
334 PARAMETER NCP=50
335 COMMON/COUPLE/ NCOUPL(NCP,8),NCPLG
336 DIMENSION NODCPL(NN),NODINT(NN)
337 C-----NODCPL = -1 NODE UNCOUPLED
338 C-----NODCPL = 0 SLAVE NODE
339 C-----NODCPL = NCOUPL(1,J) MASTER NODE
340 C-----
341 DO 5 I=1,NN
342 NODCPL(I)=-1
343 PRINT 200
344 C-----
345 READ 100,NCPLG
346 C-----
347 IF(NCPLG.EQ.0) GOTO 70
348 PRINT 205
349 DO 20 I=1,NCPLG
350 C-----
351 READ 100,(NCOUPL(I,J),J=1,8)
352 C-----
353 II=NCOUPL(I,1)
354 C-----COUPLED NODES ARE ALWAYS INTERFACE NODES
355 NODCPL(II)=I
356 DO 10 J=2,8
357 II=NCOUPL(I,J)
358 IF(II.EQ.0) GOTO 10
359 NODINT(II)=0
360 NODCPL(II)=I
361 JJ=J
362 1 CONTINUE
363 PRINT 210,(NCOUPL(I,J),J=1,JJ)
364 2 CONTINUE
365 RETURN
366 3 CONTINUE
367 PRINT 220
368 100 FORMAT()
369 200 FORMAT(//' COUPLED NODES'/1X,13(1H*))
370 205 FORMAT(' MASTER SLAVES')
371 210 FORMAT(I3,6X,7I3)
372 220 FORMAT(' NO COUPLED NODES')
373 RETURN
374 END
375
376 SUBROUTINE COUPLB(V)
377 C-----ADD SLAVE NODE QUANTITIES TO MASTER NODE QUANTITIES
378 C-----FOR EACH COUPLING NODE GROUP
379 DIMENSION V(1)
380 PARAMETER NCP=50
381 COMMON /COUPLE/ NCOUPL(NCP,8),NCPLG
382 C-----
383 IF(NCPLG.EQ.0)RETURN

```

```

384      DO 30 I=1,NCPLG
385      MNOD=NCOUPL(I,1)
386      VMN=V(MNOD)
387      DO 10 J=2,8
388      NOD=NCOUPL(I,J)
389      IF(NOD.EQ.0) GOTO 20
390      VMN=VMN+V(NOD)
391      10 CONTINUE
392      C-----
393      20 CONTINUE
394      V(MNOD)=VMN
395      30 CONTINUE
396      RETURN
397      END
398
399      SUBROUTINE COUPLC(T)
400      C-----UPDATE SLAVE NODE TEMPERATURE
401      DIMENSION T(1)
402      PARAMETER NCP=50
403      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
404      IF(NCPLG.EQ.0) RETURN
405      DO 40 I=1,NCPLG
406      MNOD=NCOUPL(I,1)
407      TMNOD=T(MNOD)
408      DO 10 J=2,8
409      NOD=NCOUPL(I,J)
410      IF(NOD.EQ.0) GOTO 20
411      T(NOD)=TMNOD
412      10 CONTINUE
413      RETURN
414      END
415
416      SUBROUTINE CTEMP(NODE,T,P,EN,FLOW,F,DELT,I,NODEL,MNODEL,N,EV4,NDC)
417      C-----CALCULATE TEMPERATURE OF INTERFACE NODES
418      PARAMETER NCP=50
419      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
420      DIMENSION P(1),EN(1),FLOW(1),F(1),NODEL(4,1),MNODEL(1),N(1),EV4(1)
421      DATA EPS/,CC5/
422      EN(NODE)=EN(NODE)+(FLOW(NODE)-F(NODE))*DELT
423      ENI=EN(NODE)
424      PI=P(NODE)
425      DO 40 J=1,5
426      T=ENI/PI
427      P1=0.
428      CALL INTP(MNODEL(NODE),NODEL(1,NODE),N,EV4,T,P1)
429      IF (NDC.LT.0) GOTO 30
430      DO 20 I=2,8
431      NOD=NCOUPL(NDC,I)
432      IF (NOD.EQ.0) GOTO 30
433      CALL INTP(MNODEL(NOD),NODEL(1,NOD),N,EV4,T,P1)
434      20 ERR=(PI-P1)/(PI+P1)
435      IF (ABS(ERR).LT.EPS) GOTO 50
436      PI=(P1+PI)/2.
437      40 CONTINUE
438      PRINT 200,NODE,T,ERR
439      200 FORMAT(' CONVERGENCE NOT ACHIEVED FOR NODE',I5,' TEMP=',G9.3,
440      1 ' ERR=',G9.2)
441      50 T=ENI/P1
442      P(NODE)=P1
443      RETURN
444      END
445
446      SUBROUTINE DTIME(NN,P,DTA,MAX,NODINT,NODCPL,TIME,DELT,I,NODT)
447      C-----THIS ROUTINE CALCULATES TIME INCREMENT

```



```

442      DIMENSION P(NN),NODINT(NN),NODCPL(NN),DTA(NN)
443      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KT*AX,KUPDA
444      DELTI=DTMAX
445      DO 10 I=1,NN
446      IF (NODCPL(I).EQ.0) GOTO 10
447      IF (NODINT(I).LT.0) GOTO 10
448      DUM=TIMFAC*P(I)/DTA(I)
449      DELTI=AMIN1(DELTI,DUM)
450      IF (DELTI.EQ.DUM) NODT=I
451 10    CONTINUE
452      DUM=TOUT(II)-TIME
453      IF (DUM*CT.^) DELTI=AMIN1(DELTI,DUM)
454      RETURN
455      END
456
457      SUBROUTINE ENCLOS(X,Y)
458  C-----THIS IS THE FIRST OF A SET OF ROUTINES FOR CALCULATION OF
459  C-----THE RATE OF CONVECTION AND RADIATION HEAT EXCHANGE IN VOIDS
460  C-----IMBEDDED IN SOLIDS. THE SAME SURFACE NODES AS FOR THE SOLID STATE
461  C-----FINITE ELEMENT ANALYSIS ARE EMPLOYED.
462  C-----
463  C-----PROGRAMMED BY
464  C-----ULF WICKSTROM
465  C-----JUNE 1977
466  C-----REVISED FEB 1979
467  C-----
468  C-----MAJOR VARIABLES,
469  C-----      NUMB - NUMBER OF NODES IN THE NODE GROUPS
470  C-----      NBOUND - NODE NUMBERS IN THE NODE GROUPS
471  C-----      SAREA - AREA BETWEEN NODES. THIRD DIMENSION ASSUMED UNITY
472  C-----      NENC - NUMBER OF VOIDS
473  C-----      NENCNG - NUMBER OF NODE GROUPS SURROUNDING EACH VOID
474  C-----      IGREN - NODE GROUP NUMBERS SURROUNDING EACH VOID
475  C-----      NNODEN - NUMBER OF NODES SURROUNDING EACH VOID
476  C-----      INODEN - ALL NODE NUMBERS IN ALL VOIDS
477  C-----      E - NODE RADIATION MATRICES
478  C-----      EPSG - EMISSIVITY OF NODE GROUP ZONES
479  C-----      H - CONVECTION VECTORS
480  C-----      BETA - CONVECTION FACTORS OF THE NODE GROUP ZONES
481  C-----      TAIR - VOID AIR TEMPERATURE
482  C-----
483      PARAMETER NB=10,NNB=30,NNB2=2*NNB
484      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),SAREA(NB,NNB),
485      1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
486      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
487      1 XSYM(2),YSYM(2)
488      COMMON/ENRAD/E(1000)
489      COMMON/ENCON/H(57),TAIR(2)
490      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
491      COMMON/DIM/MAXNG,MAXNOD
492      DATA MAXNG,MAXNOD/4,25/
493      LOGICAL LEN
494      LOGICAL XSYM,YSYM,SYM,LDUM
495      INTEGER EN
496      PRINT 100
497  C-----READ CONTROL CARD
498  C-----IF NO VOIDS IN STRUCTURE RETURN
499      READ 90,CONTRO
500      IF (CONTRO.EQ.4HVOID) GOTO 10
501      PRINT 100
502      RETURN
503 10    CONTINUE
504      TAIR=TINIT
505      LEN=.TRUE.

```

```

51  C-----READ AND ESTABLISH NODE GROUP DATA
52  C-----READ NUMBER OF VOIDS
53  READ 100,NENC
54  PRINT 200,NENC
55  IND=0
56  C-----EACH VOID
57  DO 150 EN=1,NENC
58  C-----READ SYMMETRI PROPERTIES AND NODE GROUPS THAT DEFINES THE
59  C-----V.I.
60  READ 100,XSYM(EN),YSYM(EN),(IGREN(EN,J),J=1,MAXNG)
61  SYM=XSYM(EN).OR.YSYM(EN)
62  IND=IND+1
63  I1=IGREN(EN,1)
64  INODEN(IND)=NBOUND(I1,1)
65  NODE1=INODEN(IND)
66  C-----EACH NODE GROUPE
67  DO 20 IG=1,MAXNG
68  I1=IGREN(EN,IG)
69  IF(I1.EQ.0) GOTO 30
70  I1=I1
71  NODI=NODEN(I1)
72  LDUM=INODEN(IND).NE.NBOUND(I1,1)
73  IF(LDUM) PRINT 500,EN,IND,I1
74  IF(LDUM) STOP
75  C-----EACH ZONE
76  DO 20 I=2,NUM1
77  IND=IND+1
78  INODEN(IND)=NBOUND(I1,I)
79  CONTINUE
80  PRINT 210,EN,(IGREN(EN,J),J=1,K1)
81  IF(XSYM(EN)) PRINT 250,EN
82  IF(YSYM(EN)) PRINT 260,EN
83  NENCNG(EN)=K1
84  IF(SYM) GOTO 40
85  LDUM=INODEN(IND).NE.NODE1
86  IF(LDUM) PRINT 510,EN,NODE1,INODEN(IND)
87  IF(LDUM) STOP
88  IND=IND+1
89  CONTINUE
90  VNODEN(EN)=IND
91  CONTINUE
92  CALL FNRRAD1(X,Y)
93  CALL ENCON1
94  RETURN
95  C-----
96  FORMAT(A4)
97  C-----
98  FORMAT(I)
99  C-----
100  FORMAT(//' VOIDS'//' *****')
101  C-----
102  FORMAT(//' THIS STRUCTURE HAS NO VOIDS'//)
103  C-----
104  FORMAT(//' NUMBER OF VOIDS=',I2//)
105  C-----
106  FORMAT(' VOID NUMBER',I2,' IS SURROUNDED BY THE FOLLOWING '
107  1 ' ,NODE GROUP(S)',4I3)
108  C-----
109  FORMAT(' VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE X-AXIS
110  1 ')
111  C-----
112  FORMAT(' VOID NUMBER',I2,' IS SYMMETRICAL AROUND THE Y-AXIS
113  1 ')
114  C-----
115  FORMAT(///' SURROUNDING NODEGROUPS NOT COMPATIBLE'//' EN=',I3,
116  1 ' IND=',I3,' I1=',I3)
117  C-----
118  FORMAT(///' FIRST AND LAST NODE ARE NOT IDENTICAL FOR '
119  1 ' ,VOID NUMBER',I3//' FIRST NODE=',I4//' SECOND NODE=',I4)
120  END
121  C-----
122  SUBROUTINE ENCLOS2(T,FLOW)
123  C-----THIS ROUTINE IS CALLED FROM THE BASIC FINITE ELEMENT PROGRAM
124  C-----TO CALCULATE THE RATE OF HEAT EXCHANGE BETWEEN ENCLOSURE SURFACES

```

```

576 C-----AS AFUNCTION OF CURRENT TEMPERATURE
577 DIMENSION T(1),FLOW(1)
578 COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
579 1 XSYM(2),YSYM(2)
580 LOGICAL LEN
581 IF(.NOT.LEN) RETURN
582 C-----CALCULATE RATE OF RADIATION HEAT EXCHANGE
583 CALL ENRAD2(T,FLOW)
584 C-----CALCULATE RATE OF CONVECTION HEAT EXCHANGE
585 CALL ENCON2(T,FLOW)
586 RETURN
587 END
588
589 SUBROUTINE ENCON1
590 C-----THIS ROUTINE FORMS CONVECTION ARRAY H
591 PARAMETER NB=10,NB2=30,NNB2=2*NNB
592 COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
593 1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
594 COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
595 1 XSYM(2),YSYM(2)
596 COMMON/ENCON/H(50),TAIR(2)
597 COMMON/DUMMY/HZ(25),DUM2(25)
598 LOGICAL LEN
599 LOGICAL XSYM,YSYM,SYM
600 INTEGER EN
601 IND=1
602 C-----FORM ZONE CONVECTION ARRAY
603 C-----EACH VOID
604 DO 150 EN=1,NENC
605 SYM=XSYM(EN).OR.YSYM(EN)
606 IN=0
607 NENG=NENCNG(EN)
608 C-----EACH NODE GROUP
609 DO 10 IG=1,NENG
610 I1=IGREN(EN,IG)
611 NUMI=NUMB(I1)
612 BE=BETA(I1)
613 C-----EACH ZONE
614 DO 10 I=2,NUMI
615 IN=IN+1
616 HZ(IN)=BE*BAREA(I1,I)
617 C-----FORM NODE CONVECTION ARRAY
618 CALL HTRANS(HZ,H(IND),IN,SYM)
619 N=IN
620 IF(SYM) N=N+1
621 IND=IND+N
622 150 CONTINUE
623 RETURN
624 END
625
626 SUBROUTINE ENCON2(T,FLOW)
627 C-----THIS ROUTINE CALCULATES THE AIR TEMPERATURE AND CONVECTIVE HEAT
628 C-----FLOW IN EACH ENCLOSURE
629 DIMENSION T(1),FLOW(1)
630 COMMON/ENCON/H(50),TAIR(2)
631 COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
632 1 XSYM(2),YSYM(2)
633 COMMON/DUMMY/HBAR(25),TEN(25)
634 LOGICAL LEN
635 INTEGER EN
636 DATA PE/.CCG1/
637 IND=0
638 C-----EACH VOID
639 DO 150 EN=1,NENC

```

```

540      N=NNODEN(EN)
541      C-----STORE APPROPRIATE NODAL TEMPERATURES IN DUMMY VECTOR TEN
542      DO 10 I=1,N
543          NODE=INODEN(IND+I)
544          TEN(I)=T(NODE)
545      C-----CALCULATE THE AIR TEMPERATURE TA BY ITERATION
546      C-----USE STARTING VALUE FROM FORMER TIME STEP
547      TA=TAIR(EN)
548      DO 50 ITER=1,10
549          SHBAR=0.
550          SHBT=0.
551      C-----EACH NODE
552      DO 20 I=1,N
553          TDIF=ABS(TEN(I)-TA)
554          IF (TDIF.LT..0001) HBAR(I)=0.
555          IF (TDIF.LT..0001) GOTO 20
556          HBAR(I)=H(I)*TDIF**.73
557          SHBAR=SHBAR+HBAR(I)
558          SHBT=SHBT+HBAR(I)*TEN(I)
559      20      CONTINUE
560          IF (SHBAR) 25,90,25
561      25      TANEW=SHBT/SHBAR
562      C-----CONVERGENCE CHECK OF AIR TEMPERATURE
563          IF (ABS((TANEW-TA)/(TANEW+TA)).LT.PE) GOTO 60
564          IF (ITER.GT.10) GOTO 30
565          TX=TA
566          TY=TANEW
567          TA=(TANEW+TA)*.5
568          GOTO 50
569      30      DX=TX-TA
570          DY=TY-TANEW
571          D=DY-DX
572          IF (D) 40,70,40
573      40      DN=TX*DY-TY*DX
574          TX=TA
575          TY=TANEW
576      C-----USE LINEAR INTERPOLLATION TO SPEED UP CONVERGENCE
577          TA=DN/D
578      50      CONTINUE
579          PRINT 200,EN
580          STOP
581      60      CONTINUE
582          DX=TX-TA
583          DY=TY-TANEW
584          D=DY-DX
585          IF (D) 65,70,65
586      C-----USE LINEAR INTERPOLLATION TO IMPROVE THE CALCULATED TEMPERATURE
587      65      TA=(TX*DY-TY*DX)/D
588      70      TAIR(EN)=TA
589      C-----CALCULATE CONVECTION HEAT FLOW AND ADD TO THE GLOBAL HEAT FLOW
590      C-----VECTOR FLOW
591      QTOT=0.
592      DO 80 I=1,N
593          NODE=INODEN(IND+I)
594          Q=HBAR(I)*(TA-TEN(I))
595          QTOT=QTOT+Q
596      80      FLOW(NODE)=FLOW(NODE)+Q
597      90      CONTINUE
598      CC      PRINT 220,TA,QTOT
599      IND=IND+N
700      150      CONTINUE
701      RETURN
702      200      FORMAT('///' CONVERGENCE NOT ACHIEVED FOR THE AIR TEMPERATURE',
703      1          ' IN ENCLOSURE NUMBER',I3)

```

```

704      FORMAT(1H+,50X,'AIR TEMPERATURE',F7.1,5X,'TOTAL CONVECTIVE HEAT',
705      1      'EXCHANGE',E10.3)
706      END
707
708      SUBROUTINE ENRAD1(X,Y)
709      C-----FORM RADIATION MATRICES FOR EACH VOID AND STORE THEM IN
710      C-----THE VECTOR E.
711      C-----CALCULATE VIEW-FACTOR MATRIX VIEW AND ZONE AREA VECTOR D
712      DIMENSION X(1),Y(1),A(25,25),B(25,25)
713      PARAMETER NB=10,NNB=30,NNB2=2*NNB
714      COMMON/BNOD/NJMB(NB),NBOUND(NB,NNB),SAREA(NB,NNB),
715      1      EPSG(NB),BETA(NB),CPC(NB),Fm(NB)
716      COMMON/ENCLOS/LEV,NENC,NFENG(2),IGREN(2,4),NNODEN(2),INODEN(100),
717      1      XSYM(2),YSYM(2)
718      COMMON/ENRAD/E(1000)
719      COMMON/UNIT/SIGMA,TAGS
720      COMMON/DIM/MAXNG,MAXNOD
721      COMMON/DUMMY/D(25),DUM2(25)
722      DIMENSION VIEW(25,25)
723      EQUIVALENCE (A(1),VIEW(1))
724      DATA IND,IE/D,1/
725      LOGICAL LEN
726      LOGICAL XSYM,YSYM,SYM
727      INTEGER EN
728      C-----EACH VOID
729      DO 150 EN=1,NENC
730      CALL VIEWFC(X,Y,D,EN,VIEW,MAXNOD)
731      C-----FORM THE MATRICES A AND B
732      NENG=NENG(EN)
733      IN=0
734      C-----EACH NODE GROUP
735      DO 120 IG=1,NENG
736      I1=IGREN(EN,IG)
737      NJ*I=NUMI(I1)
738      C-----EACH ZONE
739      DO 120 I=2,NUMI
740      IN=IN+1
741      JN=0
742      DO 120 JG=1,NENG
743      J1=IGREN(EN,JG)
744      NUMJ=NUMJ(J1)
745      EPSJ=EPSG(J1)
746      DO 120 J=2,NUMJ
747      JA=JN+1
748      B(IN,JN)=VIEW(IN,JN)*SIGMA
749      A(IN,JN)=-VIEW(IN,JN)*(1.-EPSJ)/EPSJ/D(JN)
750      IF(IN.NE.JN) GOTO 120
751      B(IN,JN)=-SIGMA+B(IN,JN)
752      A(IN,JN)=1./EPSJ/D(JN)+A(IN,JN)
753      CONTINUE
754      N=IN
755      C-----INVERT A AND STORE RESULT IN A
756      CALL INVER(A,N,MAXNOD)
757      C-----MULTIPLY A AND B AND STORE RESULT IN A
758      CALL MULT(A,B,N,MAXNOD)
759      SYM=.FALSE.
760      IF(XSYM(EN).OR.YSYM(EN)) SYM=.TRUE.
761      NZ=N
762      IF(SYM) N=N+1
763      C-----TRANSFORM THE LOCAL RADIATION MATRICE A AND STORE THE RESULT IN
764      C-----VECTOR E
765      C-----B IS EMPLOYED AS A DUMMY MATRIX
766      CALL ETRANS(A,B,E(IE),N,NZ,SYM,MAXNOD)
767      IE=IE+SYM

```

```

744      IND=IND+N
745      CONTINUE
746      RETURN
747      END

748      SUBROUTINE ENRAD2(T,FLOW)
749      C-----THIS ROUTINE CALCULATES THE RADIATION HEAT FLOW TO EACH NODE OF A
750      C-----ENCLOSURE SURFACE AND ADDS THE RESULT TO THE GLOBAL HEAT FLOW
751      C-----VECTOR FLOW
752      DIMENSION T(1),FLOW(1)
753      COMMON/ENCLUS/LEN,NENC,NENCLG(2),IGREN(2,4),NNODEN(2),INODEN(100),
754      1 XSYM(2),YSYM(2)
755      COMMON/ENRAD/E(1000)
756      COMMON/UNIT/SIGMA,TABS
757      COMMON/DUMMY/ETA(25),Q(25)
758      LOGICAL LEN
759      INTEGER EN
760      IE=1
761      IND=1
762      C-----EACH NODE
763      DO 150 EN=1,NENC
764      N=NNODEN(EN)
765      C-----CALCULATE ABSOLUTE TEMPERATURES TO THE FOURTH POWER FOR FOR THE
766      C-----NODES OF THE ENCLOSURE SURFACE
767      DO 10 I=1,N
768      NCDE=INODEN(IND+I)
769      DUM=T(NCDE)+TABS
770      DUM=DUM*DUM
771      ETA(I)=DUM*DUM
772      C-----CALCULATE ENCLOSURE SURFACE RADIATION HEAT EXCHANGE VECTOR Q=E*ETA
773      CALL RADVEC(E(IE),ETA,N,Q)
774      C-----ADD TO GLOBAL HEAT FLOW VECTOR FLOW
775      DO 20 I=1,N
776      NODE=INODEN(IND+I)
777      20 FLOW(NODE)=FLOW(NODE)+Q(I)
778      IE=IE+N*N
779      IND=IND+N
780      150 CONTINUE
781      RETURN
782      END

783      SUBROUTINE ETRANS(A,B,E,N,NZ,SYM,MAX)
784      C-----THIS ROUTINE TRANSFORMS THE ZONE RADIATION MATRIX A TO A NODE
785      C-----RADIATION MATRIX AND STORE THE RESULT IN E
786      C-----IF SYMMETRY IS PRESENT EXPAND RADIATION MATRIX
787      DIMENSION E(N,N),A(MAX,MAX),B(MAX,MAX)
788      LOGICAL SYM
789      C-----E=SAT*A*SA
790      C-----S=SAT*A
791      DO 10 I=2,NZ
792      DO 10 J=1,NZ
793      10 B(I,J)=A(I-1,J)+A(I,J)
794      DO 30 J=1,NZ
795      IF(SYM) GOTO 20
796      B(1,J)=A(1,J)+A(NZ,J)
797      GOTO 30
798      20 B(1,J)=A(1,J)
799      B(N,J)=A(NZ,J)
800      30 CONTINUE
801      C-----E=B*SA
802      DO 50 I=1,N
803      DO 50 J=2,NZ
804      50 E(I,J)=.25*(B(I,J)+B(I,J-1))
805      DO 70 I=1,N

```



```

832         IF(SYM) GOTO 60
833         F(I,1) = .25 * (F(I,1) + F(I,4))
834         GOTO 70
835     60     E(I,1) = .25 * B(I,1)
836         E(I,4) = .25 * B(I,4)
837     70     CONTINUE
838         RETURN
839         END
840
841     SUBROUTINE FEW2(IX,IY,NN,NE,NR,V,KTOP,X,Y,T,TT,TMAX,ELA,EV4,A,MAX,
842     1     P,W,EN,F,FLOW,AXIAL,NODCPL,NODINT,NODEL,MNODEL,DTA)
843     C-----THIS ROUTINE INITIALIZES SYSTEM ARRAYS AND
844     C-----CONTROLS TIME INTEGRATION
845     DIMENSION N(NE),KTOP(4,NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),
846     1     ELA(4,NE),EV4(NE),A(NN,MAX),P(NN),W(NN),EN(NN),F(NN),FLOW(NN)
847     2     ,NODCPL(NN),NODINT(NN),NODEL(4,NN),MNODEL(NN),DTA(NN)
848     PARAMETER MNV=20,MNR=10
849     COMMON/FIRE/TIM(SC),TE(SC),TITFIR
850     COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR),
851     COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
852     LOGICAL TMAX,AXIAL
853     LOGICAL FIN,CON,NODI,UPDA
854     DATA FIN/.FALSE./
855
856     5     CONTINUE
857     KTIME=0
858     CON=.TRUE.
859     C-----INITIALIZE NODAL TEMPERATURES
860     CALL INIT(NN,T,TT,TMAX,NODINT)
861     C-----INPUT FIRE BOUNDARY TEMPERATURE
862     CALL BFIRE(FIN)
863     C-----FIRST TIME INCREMENT FOR CALCULATING INCREMENT LENGTH ONLY
864     DELTI=0.
865     C-----IF FIN=.TRUE. ANALYZE NEW FIRE
866     C-----IF FIN=.TRUE. TERMINATE RUN
867     IF(FIN)GOTO 1000
868     CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT,
869     1     AXIAL)
870     C-----INITIALIZE NODAL ENTHALPY VECTOR EN
871     C-----HOMOGENEOUS NODES - EN = ENTHALPY(HEAT) PER UNIT VOLUME
872     C-----INTERFACE NODES - EN = ENTHALPY(HEAT)
873     DO 10 I=1,NN
874         NDI=NODINT(I)
875         IF(NODCPL(I).EQ.0.OR.NDI.LT.0) GOTO 10
876         IF(NDI.GT.0) CALL XVERSY(TE,ENT,MNV,NDI,T(I),EN(I))
877     C-----MASTER NODES AND INTERFACE NODES
878         IF(NODCPL(I).GT.0.AND.NDI.EQ.0) CALL MINTP(I,NODCPL(I),P)
879         IF(NDI.EQ.0) EN(I)=P(I)*T(I)
880     10     CONTINUE
881
882     C-----
883     C-----START TIME INTEGRATION LOOP
884     C-----
885     700     CONTINUE
886     DUM1=FLOAT(KTIME)/FLOAT(KUPDA)
887     DUM2=AINT(DUM1)
888     UPDA=DUM1.EQ.DUM2.OR.KTIME.EQ.1
889     KTIME=KTIME+1
890     C-----CALCULATE INTERNAL HEAT FLOW BY CONDUCTION
891     IF(UPDA)
892     1     CALL ASSA2(NN,NE,N,KTOP,X,Y,ELA,T,TT,TMAX,AXIAL,MAX,A)
893     DO 20 I=1,NN
894     20     DTA(I)=A(I,MAX)
895     CALL MPACKV(A,T,F,MAX,NN)
896     C-----GET FIRE TEMPRATURE

```

```

896      CALL XVERSY(TIM,TD,57,1,TIME,TFIRE)
897      C-----CALCULATE BOUNDARY HEAT FLOW
898      CALL FQBND(B,T,FLOW,DTA,NN,MAX,TFIRE)
899      C-----CALCULATE INTERNALLY GENERATED HEAT FLOW
900      CALL FQGEN(NN,NE,N,KTOP,EV4,T,FLOW)
901      C-----CALCULATE ENCLOSURE (VOID) HEAT FLOW
902      CALL ENCLOS2(T,FLOW)
903      C-----CALCULATE HEAT CAPACITY MATRIX AT CURRENT TEMPERATURE
904      CALL ASSP2(NN,N,X,Y,T,TT,TMAX,EV4,NODEL,MNODEL,P,W,NODINT,
905      1 AXIAL)
906      C-----SUM APPROPRIATE QUANTITIES OF COUPLED NODES
907      CALL COUPLB(F)
908      CALL COUPLR(FLOW)
909      CALL ACOUPL(A,DTA,NN,MAX)
910      C-----CALCULATE NEW NODAL TEMPERATURES
911      DO 50 I=1,NN
912      NDC=NODCPL(I)
913      NDI=NODINT(I)
914      IF(NDI.LT.0.OR.NDC.EQ.0) GOTO 50
915      C-----HOMOGENEOUS NODES ONLY
916      NDI=NDI.GT.0
917      IF(NDI)CALL HTEMP(T(I),W(I),EN(I),FLOW(I),F(I),NDI,DELTI)
918      IF(NDI) GOTO 50
919      C-----INTERFACE NODES
920      IF(NODCPL(I).GT.0) CALL MINTP(I,NODCPL(I),P)
921      CALL CTEMP(I,T(I),P,EN,FLOW,F,DELT,NODEL,MNODEL,N,EV4,NDC)
922      50 CONTINUE
923      C-----SET PRESCRIBED NODAL TEMPERATURES
924      CALL PTBND(B,T,TFIRE)
925      CALL COUPLC(T)
926      C-----PRINT CURRENT NODAL AND VOID AIR TEMPERATURES
927      CALL OUT2(IX,IY,NN,NE,X,Y,TIME,KTIME,DELT,T,TT,TMAX,FLOW,TFIRE,
928      1 NODT,AXIAL)
929      C-----SET CON=.FALSE. TO TERMINATE TIME INTEGRATION
930      CALL MAXCO(NN,TMAX,TT,T,TIME,KTIME,CON)
931      C-----CALCULATE NEW TIME INCREMENT DELTI
932      CALL DTIME(NN,P,DTA,MAX,NODINT,NODCPL,TIME,DELT,NODT)
933      TIME=TIME+DELT
934      IF(CON) GOTO 700
935      C-----
936      C-----END TIME INTEGRATION LOOP
937      C-----
938      720 CONTINUE
939      TIME=TIME-DELT
940      C-----PRINT MAXIMUM TEMPERATURE OBTAINED DURING ANALYSIS
941      CALL OUTMA2(IX,IY,NN,NE,X,Y,TIME,KTIME,T,TT,TMAX,FLOW,AXIAL)
942      GOTO 5
943      1000 RETURN
944      END
945
946      SUBROUTINE FQBND
947      C-----THIS ROUTINE FORMS RADIATION AND CONVECTION MATRICES BR AND BC
948      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
949      PARAMETER NB=10,NNB=30,NNB2=2*NNB
950      COMMON/FQB/NFQNG,NFQNG(NB),TR(NNB),TC(NNB)
951      1 BR(NNB2),BC(NNB2)
952      COMMON/ENOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
953      1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
954      LOGICAL FA,FA1
955      C-----READ NUMBER OF BOUNDARY NODES GROUPS
956      READ 100,NFQNG
957      C-----
958      IF(NFQNG.EQ.0) RETURN
959      PRINT 200

```

```

960      IND=1
961      C-----EACH BOUNDARY FLOW NODE GROUP
962      DO 10 IB=1,NFGNG
963      C-----
964      READ 100,FA1,ING1
965      C-----IF FA1=.TRUE. FIRE BOUNDARY ELSE AMBIENT TEMPERATURE
966      C-----ING1 = NODE GROUP NUMBER
967      FA(ING1)=FA1
968      VFQG(IB)=ING1
969      NUMI=NUMB(ING1)
970      BET=BETA(ING1)
971      EPSIG=EPSG(ING1)*SIGMA
972      IF(EPSIG.EQ.0.AND.BET.EQ.0) PRINT 300
973      IF(EPSIG.EQ.0.AND.BET.EQ.0) STOP
974      CALL BRBCA(BR(IND),BC(IND),EPSIG,BET,BAREA,NUMI,NB,ING1)
975      IND=IND+2*NUMI
976      IF(FA1) PRINT 210,ING1
977      IF(.NOT.FA1) PRINT 220,ING1
978      10 CONTINUE
979      100 FORMAT()
980      200 FORMAT('!' PRESCRIBED FLOW BOUNDARY'/1X,24(1H*)/
981      1 /' NODE GROUPS AND TYPES OF BOUNDARIES'//)
982      210 FORMAT('' NODE GROUP',I3,' FIRE BOUNDARY')
983      220 FORMAT('' NODE GROUP',I3,' AMBIENT BOUNDARY')
984      300 FORMAT('' BOTH EMISSIVITY AND CONVECTION FACTOR ZERO')
985      RETURN
986      END
987
988      SUBROUTINE FQBNDB(T,FLOW,DTA,NN,MAX,TFIRE)
989      C-----THIS ROUTINE PREPARES CALCULATION OF PRESCRIBED BOUNDARY FLOW
990      DIMENSION T(NN),DTA(NN),FLOW(NN)
991      PARAMETER NB=10,NNB=30,NNB2=2*NNB
992      COMMON/FQB/NFGNG,VFQG(NB),TR(NNB),TC(NNB)
993      1 ,BR(NNB2),BC(NNB2),TRD(NNB),TCD(NNB)
994      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
995      1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
996      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
997      LOGICAL FA
998      C-----NULL FLOW VECTOR
999      DO 777 I=1,NN
1000      777 FLOW(I)=0.
1001      C-----RETURN IF NO PRESCRIBED BOUNDARY FLOW
1002      IF(NFGNG.EQ.0) RETURN
1003      TF4=(TFIRE+TABS)**4
1004      IND=1
1005      C-----EACH BOUNDARY FLOW NODE GROUP
1006      DO 30 IB=1,NFGNG
1007      TG4=TAMB4
1008      TG=TAMB
1009      ING1=VFQG(IB)
1010      IF(FA(ING1)) TG=TFIRE
1011      IF(FA(ING1)) TG4=TF4
1012      NUMI=NUMB(ING1)
1013      CP=CPG(ING1)
1014      DO 20 I=1,NUMI
1015      NODE=NBOUND(ING1,I)
1016      TNODE=T(NODE)
1017      TNABS=TNODE+TABS
1018      C-----RADIATION
1019      TPD(I)=4.*TNABS**3
1020      TR(I)=TG4-TNABS**4
1021      C-----CONVECTION
1022      DUM=TG-TNODE
1023      TCD(I)=CP*ABS(DUM)**(CP-1.)

```

```

1024      IF(DUM) 5,20,10
1025      5      TC(I)=-(-DUM)**CD
1026      GOTO 20
1027      10     TC(I)=DUM**CP
1028      20     CONTINUE
1029      C-----
1030      CALL BRBCB(BR(IND),BC(IND),TR,TC,TRD,TRD,NUMI,DTA,NN,
1031      1      MAX,FLOW,TG,T,ING1)
1032      IND=IND+2*NUMI
1033      30     CONTINUE
1034      RETURN
1035      END
1036
1037      SUBROUTINE FGGEN(NN,NE,N,KTOP,EV4,T,FLOW)
1038      C-----CALCULATE INTERNALLY GENERATED HEAT
1039      DIMENSION N(NE),KTOP(4,NE),EV4(NE),T(NN),FLOW(NN)
1040      PARAMETER MNV=20,MNR=10
1041      COMMON/RMAT/CCC(MNR),TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),
1042      1      ENT(MNV,MNR),CR(MNV,MNR),TQ(MNV,MNR),QE(MNV,MNR),LQ(MNR)
1043      LOGICAL CCC,LQ
1044      DO 20 I=1,NE
1045      N1=N(I)
1046      IF(.NOT.LQ(N1)) GOTO 20
1047      DO 10 K=1,4
1048      NOD=KTOP(K,I)
1049      CALL XVERSY(TQ,QE,MNV,N1,T(NOD),FGEN)
1050      10     FLOW(NOD)=FLOW(NOD)+EV4(I)*FGEN
1051      20     CONTINUE
1052      RETURN
1053      END
1054
1055      SUBROUTINE GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,EV4)
1056      C-----THIS SUBROUTINE COMPUTES ELEMENT GEOMETRICAL CONSTANTS
1057      C-----
1058      C-----      NN      NUMBER OF NODES
1059      C-----      NE      NUMBER OF ELEMENTS
1060      C-----      N      ELEMENT REGION NUMBER
1061      C-----      X,Y     NODE COORDINATES
1062      C-----      AXIAL   TRUE IF AXIAL SYMMETRIC PROBLEM
1063      C-----      ELA     ELEMENT GEOMETRIC CONSTANTS
1064      C-----      EV4     ELEMENT VOLUME/4
1065      DIMENSION X(NN),Y(NN),EV4(NE),ELA(4,NE),KTOP(4,NE),N(NE)
1066      PARAMETER MNR=10
1067      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
1068      LOGICAL AXIAL,ELFICT
1069      DO 5 I=1,NE
1070      N1=N(I)
1071      K1=KTOP(1,I)
1072      K4=KTOP(4,I)
1073      A=X(K4)-X(K1)
1074      B=Y(K4)-Y(K1)
1075      ET1=ET(N1)
1076      ELA(1,I)=ET1*(A*A+B*B)/3./A/B
1077      ELA(2,I)=ET1*(-2*A*A+B*B)/6./A/B
1078      ELA(3,I)=ET1*(A*A-2.*B*B)/6./A/B
1079      ELA(4,I)=-ET1*(A*A+B*B)/6./A/B
1080      EV4(I)=ET1*A*B/4.
1081      5      CONTINUE
1082      RETURN
1083      END
1084
1085      SUBROUTINE HTEMP(T,W,EN,FLOW,F,N1,DELT1)
1086      C-----CALCULATE TEMPERATURE FOR HOMOGENEOUS NODES
1087      PARAMETER MNV=20,MNR=10

```

```

1088      COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR),
1089      1      CR(MNV,MNR)
1090      D=(FLOW-F)*DELT
1091      EN=EN+D/W
1092      CALL XVERSY(ENT,TE,MNV,N1,EN,T)
1093      RETURN
1094      END
1095
1096      SUBROUTINE HTRANS(HZ,H,N,SYM)
1097      C-----THIS ROUTINE TRANSFORMS THE ZONE CONVECTION ARRAY H TO A NODE
1098      C-----CONVECTION ARRAY STORED IN THE SAME ARRAY
1099      C-----IF SYMMETRY IS PRESENT EXPAND CONVECTION VECTOR
1100      DIMENSION HZ(1),H(1)
1101      LOGICAL SYM
1102      DO 10 I=2,N
1103      H(I)=.5*(HZ(I-1)+HZ(I))
1104      10 CONTINUE
1105      IF(SYM) GOTO 20
1106      H(1)=.5*(HZ(N)+HZ(1))
1107      GOTO 30
1108      20 CONTINUE
1109      H(1)=.5*HZ(1)
1110      H(N+1)=.5*HZ(N)
1111      30 RETURN
1112      END
1113
1114      SUBROUTINE INIT(YN,T,TT,TMAX,NODINT)
1115      C-----SET INITIAL NODAL TEMPERATURE
1116      DIMENSION T(NN),TT(NN),TMAX(NN),NODINT(NN)
1117      LOGICAL TMAX
1118      COMMON/TOUT/II,TOUT(100),TIMMAX
1119      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
1120      II=1
1121      DO 1 I=1,NN
1122      IF(NODINT(I).LT.0) GOTO 1
1123      T(I)=TINIT
1124      TT(I)=T(I)
1125      1 TMAX(I)=.FALSE.
1126      RETURN
1127      END
1128
1129      SUBROUTINE INTER(NN,NE,NR,NX,NY,KTOP,N,NODINT,NODCPL)
1130      C-----THIS ROUTINE FORMS VECTOR FOR IDENTIFICATION OF
1131      C-----INTERFACE AND FICTITIOUS NODES
1132      C-----NODINT=-1 - FICTITIOUS NODE
1133      C-----NODINT= 0 - INTERFACE NODE
1134      C-----NODINT= 1 - HOMOGENEOUS NODE
1135      PARAMETER MNR=10,NCP=50
1136      COMMON/RGEO/ELFICT(MNR)
1137      COMMON/COUPLE/NCPL(NCP,8),NCPLG
1138      DIMENSION KTOP(4,NE),N(NE),NODINT(NN),NODCPL(NN)
1139      LOGICAL ELFICT
1140      PRINT 200
1141      DO 5 I=1,NN
1142      NODINT(I)=-1
1143      IF(NR.EQ.1) GOTO 50
1144      NX1=NX-1
1145      NY1=NY-1
1146      IF(NY.EQ.2)GOTO 25
1147      C-----
1148      DO 20 I=1,NX1
1149      INY=(I-1)*NY
1150      INY1=(I-1)*NY1
1151      IE1=INY1+1

```

```

1152      N1=N(IE1)
1153      C-----
1154      DO 20 J=2,NY1
1155      IF(ELFICT(N1)) GOTO 15
1156      DO 10 IDUM=1,3
1157      IE2=IE1+(IDUM-2)*NY1+1
1158      IF(IE2.LE.0.OR.IF2.GT.NE)GOTO 10
1159      N2=N(IE2)
1160      IF(ELFICT(N2)) GOTO 10
1161      IF(N1.EQ.N2) GOTO 10
1162      IF(IDUM.NE.2)GOTO 10
1163      NOD=INY+J
1164      NODINT(NOD)=0
1165      NOD=NOD+NY
1166      NODINT(NOD)=0
1167      C-----
1168      10 CONTINUE
1169      15 CONTINUE
1170      IE1=IE1+1
1171      N1=N(IE1)
1172      C-----
1173      20 CONTINUE
1174      25 CONTINUE
1175      C-----
1176      IF(NX.EQ.2)GOTO 50
1177      DO 40 I=1,NY1
1178      INX=I-1
1179      IE1=I
1180      N1=N(IE1)
1181      C-----
1182      DO 40 J=2,NX1
1183      C-----
1184      IE2=IE1+NY1
1185      N2=N(IE2)
1186      IF(ELFICT(N1).OR.ELFICT(N2)) GOTO 30
1187      IF(N1.EQ.N2) GOTO 30
1188      NOD=(J-1)*NY+I
1189      NODINT(NOD)=0
1190      NOD=NOD+1
1191      NODINT(NOD)=0
1192      C-----
1193      30 CONTINUE
1194      IE1=IE2
1195      N1=N2
1196      C-----
1197      40 CONTINUE
1198      50 CONTINUE
1199      C-----
1200      DO 70 I=1,NE
1201      N1=N(I)
1202      IF(ELFICT(N1)) GOTO 70
1203      DO 60 J=1,4
1204      NOD=KTOP(J,I)
1205      IF(NODINT(NOD).EQ.-1) NODINT(NOD)=N1
1206      70 CONTINUE
1207      C-----IF ONE NODE IN A COUPLED GROUP IS AN INTERFACE NODE
1208      C-----ALL NODES IN THE GROUP ARE CONSIDERED INTERFACE NODES
1209      IF (NCPLG.EQ.0) GOTO 120
1210      DO 110 I=1,NCPLG
1211      DO 95 J=1,R
1212      NOD=NCOUPL(I,J)
1213      IF(NOD.EQ.0) GOTO 110
1214      IF(NODINT(NOD).EQ.0) GOTO 90
1215      95 CONTINUE

```



```

1216      9C      CONTINUE
1217          DO 100 J=1,N
1218              NOD=NCOUPL(I,J)
1219              IF(NOD.EQ.0) GOTO 11C
1220              NODINT(NOD)=0
1221      10C      CONTINUE
1222      11C      CONTINUE
1223      12C      CONTINUE
1224          PRINT 2C5
1225          DO 3C I=1,NX
1226              II=(I-1)*NY
1227      3C      PRINT 21C,(NODINT(II+J),J=1,NY)
1228      205      FORMAT(/' -1 - FICTITIOUS NODE'/' 0 - INTERFACE NODE'/
1229              1 ' 1 - HOMOGENEOUS NODE')
1230      20C      FORMAT(/' INTERFACE NODES'/16(1H*))
1231      210      FORMAT(/1CX,5G12)
1232          RETURN
1233          END
1234
1235      SUBROUTINE INTP(MND,MODEL,N,EV4,TI,PI)
1236      C-----CALCULATE HEAT CAPACITY OF INTERFACE NODES
1237          DIMENSION MODEL(4),N(1),EV4(1)
1238          PARAMETER MNV=20,MNR=10
1239          COMMON/RMAT/TC(MNV,MNR),C(MNV,MNR),TE(MNV,MNR),ENT(MNV,MNR)
1240          DO 1C J=1,MND
1241              IE=MODEL(J)
1242              N1=N(IE)
1243              EV4IE=EV4(IE)
1244              CALL XVERSY(TE,ENT,MNV,N1,TI,ENI)
1245              CRA=ENI/TI
1246              PI=PI+EV4IE*CRA
1247      1C      CONTINUE
1248          RETURN
1249          END
1250
1251      SUBROUTINE INVER(A,M,MAX)
1252      C-----THIS ROUTINE INVERTS THE MATRIX A AND STORES THE RESULT IN THE
1253      C-----SAME MATRIX
1254          DIMENSION A(MAX,1)
1255          DO 200 N=1,M
1256              D=A(N,N)
1257              DO 100 J=1,M
1258      100      A(N,J)=-A(N,J)/D
1259              DO 150 I=1,M
1260                  IF(N.EQ.I) GOTO 150
1261              DO 140 J=1,M
1262                  IF(J.EQ.N) GOTO 140
1263                  A(I,J)=A(I,J)+A(I,N)*A(N,J)
1264      140      CONTINUE
1265              150 A(I,N)=A(I,N)/D
1266      200      A(N,N)=1./D
1267          RETURN
1268          END
1269
1270      COMMENT MAIN PROGRAM CODED IN NUALGOL FOR DYNAMIC ALLOCATION OF ARRAYS
1271          FOR INFORMATION ABOUT ARRAYS SEE SUBROUTINE PROG2;
1272      BEGIN
1273          INTEGER NN,NE,NR,IX,IY,MAX;
1274          REAL ARRAY XL,YL,XA,YA(1:100);
1275          BOOLEAN AXIAL;
1276          EXTERNAL FORTRAN PROCEDURE NET2,D1M2;
1277          NET2(XL,YL,IX,IY,NR,AXIAL,XA,YA,NN,NE,MAX);
1278          BEGIN
1279              BOOLEAN ARRAY TMAX(1:NN);

```



```

1344      PRINT 11C
1345      DO 39 K=1,NT
1346      PRINT 11S
1347      IF (C(K,I).GT..00001) PRINT 12C,TC(K,I),C(K,I)
1348      IF (CP(K,I).GT..00001) PRINT 13C,TE(K,I),ENT(K,I),CR(K,I)
1349      CONTINUE
39      IF (NQE.EQ.0) GOTO 45
1351      PRINT 14C,(TG(K,I),QE(K,I),K=1,NQE)
1352      CONTINUE
1353      2    CONTINUE
1354      RETURN
1355      1    FORMAT()
1356      95   FORMAT(//1X,'MATERIAL DATA'/1X,13(1H*))
1357      100  FORMAT(1X,'REGION NUMBER',I3)
1358      101  FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4)
1359      102  FORMAT(1H+,30X,'THICKNESS',F9.3/1X,20A4/
1360      1    ' CONDUCTIVITY IS KEPT CONSTANT AFTER REACHING MAXIMUM'
1361      2    ' TEMPERATURE')
1362      110  FORMAT(1X,'TEMP',6X,'CONDUCTIVITY',13X,'TEMP',8X,'ENTHALPHY',6X,
1363      1    'ENT/TEMP'/)
1364      115  FORMAT(1Y)
1365      120  FORMAT(1H+,F6.0,E15.4)
1366      130  FORMAT(1H+,3X,F6.0,E15.4,E15.4)
1367      140  FORMAT(1X,' INTERNALLY GENERATED HEAT'/1X,TEMP',6X,'HEAT'
1368      1    '/(1X,F6.0,E15.4))
1369      200  FORMAT(20A4)
1370      1000 PRINT 120C,NTC,NTE,NQE
1371      STOP
1372      1200 FORMAT(///'      PROGRAM TERMINATED WHEN READING MATERIAL INPUT'
1373      1    '/1X,NTC=',I3,' NTE=',I3,' NQE=',I3)
1374      END
1375
1376      SUBROUTINE MAXCO(NN,TMAX,TT,T,TIME,KTIME,CON)
1377      C-----SET CON=.FALSE. TO TERMINATE TIME INTEGRATION
1378      DIMENSION TMAX(NN),TT(NN),T(NN)
1379      LOGICAL TMAX
1380      LOGICAL CON
1381      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX
1382      IF (TIME.GE..99999*TIMMAX) CON=.FALSE.
1383      IF (KTIME.GT.KTMAX) PRINT 200,KTIME
1384      IF (KTIME.GT.KTMAX) CON=.FALSE.
1385      200  FORMAT(//' TERMINATED AT MAXIMUM NUMBER OF TIME '
1386      1    ', INCREMENTS KTIME='I5)
1387      RETURN
1388      END
1389
1390      SUBROUTINE MESH2(XL,YL,IX,IY,X,Y,KTOP)
1391      C-----THIS SUBROUTINE COMPUTES COORDINATES AND TOPOLOGY
1392      DIMENSION XL(1),YL(1),X(1),Y(1),KTOP(4,1)
1393      C-----COMPUTE X AND Y COORDINATES
1394      DO 5 I=1,IX
1395      DO 5 J=1,IY
1396      KK=J+IY*(I-1)
1397      X(KK)=XL(I)
1398      Y(KK)=YL(J)
1399      5    CONTINUE
1400      C-----COMPUTE THE TOPOLOGY MATRIX KTOP
1401      IX1=IX-1
1402      IY1=IY-1
1403      DO 10 I=1,IX1
1404      DO 10 J=1,IY1
1405      IE=IY1*(I-1)+J
1406      KTOP(1,IE)=IY*(I-1)+J
1407      KTOP(2,IE)=KTOP(1,IE)+1

```

```

1408      KTOP(3,IE)=KTOP(1,IE)+IY
1409      KTOP(4,IE)=KTOP(3,IE)+1
1410      10 CONTINUE
1411      RETURN
1412      END
1413
1414      SUBROUTINE MINTP(I,NODCPL,P)
1415      C-----SUM HEAT CAPACITY OF COUPLED NODES
1416      PARAMETER NCP=50
1417      COMMON/COUPLE/NCOUPL(NCP,8),NCPLG
1418      DIMENSION P(1)
1419      DO 30 J=2,P
1420      NOD=NCOUPL(NODCPL,J)
1421      IF(NOD.EQ.0) RETURN
1422      70 P(I)=P(I)+P(NOD)
1423      RETURN
1424      END
1425
1426      SUBROUTINE MPAKCV(A,X,R,MI,NN)
1427      C-----THIS ROUTINE MULTIPLIES BANDED AND PACKED SYMMETRIC MATRIX
1428      C-----WITH VECTOR A * X = R
1429      C-----A = MATRIX WITH DIAGONAL ELEMENTS IN RIGHT HAND SIDE COLUMN
1430      DIMENSION A(NV,MI),R(NN),X(NN)
1431      DO 3 I=1,NN
1432      R(I)=0.
1433      DO 2 J=1,MI
1434      IF((I+J-MI).GT.0)
1435      *R(I)=R(I)+A(I,J)*X(I+J-MI)
1436      IF(I.EQ.NN) GO TO 3
1437      I1=MIN((MI-1),(NN-I))
1438      DO 1 J=1,I1
1439      R(I)=R(I)+A(I+J,MI-J)*X(I+J)
1440      3 CONTINUE
1441      RETURN
1442      END
1443
1444      SUBROUTINE MULT(A,B,N,MAX)
1445      C-----THIS ROUTINE MULTIPLIES THE MATRICES A AND B AND STORE THE
1446      C-----RESULT IN A WITH CHANGED SIGNS
1447      DIMENSION A(MAX,MAX),B(MAX,MAX)
1448      COMMON/DUMMY/ETA(25),DUM2(25)
1449      DO 20 I=1,N
1450      DO 10 J=1,N
1451      ETA(J)=0.
1452      DO 10 K=1,N
1453      10 ETA(J)=ETA(J)+A(I,K)*B(K,J)
1454      DO 20 J=1,N
1455      20 A(I,J)=ETA(J)
1456      RETURN
1457      END
1458
1459      SUBROUTINE NET2(XL,YL,IX,IY,NR,AXIAL,XA,YA,NN,NE,MAX)
1460      C-----INPUT GEOMETRICAL DATA AND GENERATE LINES PARALLELL WITH AXIS
1461      C-----AND CALCULATE NUMBER OF GENERATED NODES AND ELEMENTS
1462      C-----
1463      C----- XL COORDINATES OF X-LINES
1464      C----- YL COORDINATES OF Y-LINES
1465      C----- XA COORDINATES OF SPECIFIED X-LINES
1466      C----- YA COORDINATES OF SPECIFIED Y-LINES
1467      C----- IX NUMBER OF X-LINES
1468      C----- IY NUMBER OF Y-LINES
1469      C----- NR NUMBER OF REGIONS
1470      C----- AXIAL .TRUE. IF AXI-SYMMETRIC PROBLEM
1471      C----- NN NUMBER NODES

```

```

1472 C----- NE          NUMBER OF ELEMENTS
1473 C----- NX          NUMBER OF SPECIFIED X-LINES
1474 C----- NY          NUMBER OF SPECIFIED Y-LINES
1475 C-----
1476     PARAMETER MNR=10
1477     COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
1478     LOGICAL AXIAL,ELFICT
1479     DIMENSION XL(1),YL(1),XA(1),YA(1),HEAD(20)
1480     PRINT 199
1481 C*****MACHINE DEPENDENT STATEMENT * IN LDC*LIB.SEQ * REQUIRED IN MAP ELEMENT
1482     CALL SEQ('TASEF')
1483     PRINT 200
1484 C-----INPUT TITLE OF RUN
1485     READ 110,HEAD
1486     PRINT 205,HEAD
1487 C-----
1488 C-----INPUT MAIN GEOMETRICAL DATA
1489 C-----
1490     READ 100,AXIAL,XMAX,YMAX,XBOX,YBOX,NR,NX,NY
1491     PRINT 210,XMAX,YMAX
1492     PRINT 220,XBOX,YBOX
1493     IF(NR.EQ.0) NR=1
1494     NR1=NR-1
1495     EPS= XMAX/10000.
1496 C-----
1497 C-----INPUT SUBREGION LIMITS
1498 C-----READ THE DIAGONAL COORDINATES FOR EACH SUBREGION
1499 C-----
1500     IF(NR.EQ.1) GOTO 5
1501     PRINT 230
1502     READ 100,(ELFICT(J),(SRDIAC(I,J),I=1,4),J=2,NR)
1503     PRINT 240,NR1,((SRDIAC(I,J),I=1,4),ELFICT(J),J=2,NR)
1504 5     CONTINUE
1505     IF(NX.EQ.0) GOTO 6
1506 C-----
1507 C-----INPUT SPECIFIED X - LINES
1508 C-----
1509     PRINT 250
1510     READ 100,(XA(I),I=1,NX)
1511     PRINT 260,(XA(I),I=1,NX)
1512 6     CONTINUE
1513     IF(NY.EQ.0) GOTO 7
1514 C-----INPUT SPECIFIED Y - LINES
1515     PRINT 270
1516     READ 100,(YA(I),I=1,NY)
1517     PRINT 280,(YA(I),I=1,NY)
1518 7     CONTINUE
1519 C-----
1520 C-----IF AN AXI-SYMMETRIC PROBLEM INPUT INNER RADIUS
1521 C-----
1522     NX=NX+1
1523     IF(.NOT.AXIAL) GOTO 8
1524     READ 100,XA(1)
1525     PRINT 300,XA(1)
1526 8     CONTINUE
1527 C-----
1528 C-----GENERATE X-LINES
1529 C-----
1530     XL(1)=XA(NX)
1531     DO 15 IX=2,100
1532     XL(IX)=XL(IX-1)+XBOX
1533 C-----
1534 C-----CONTROL OF SPECIFIED X-LINES
1535 C-----

```

```

1536      DO 10 I=1,NX
1537      IF(XL(IX-1).LT.(XA(I)-EPS)) XL(IX)=AMIN1(XL(IX),XA(I))
1538 10    CONTINUE
1539 C-----
1540 C-----CONTROL OF SUBREGION LIMITS
1541 C-----
1542      IF(NR.EQ.1) GOTO 12
1543      DO 11 I=2,NR
1544      IF(XL(IX-1).LT.(SRDIAC(1,I)-EPS)) XL(IX)=AMIN1(XL(IX),SRDIAC(1,I))
1545      IF(XL(IX-1).LT.(SRDIAC(3,I)-EPS)) XL(IX)=AMIN1(XL(IX),SRDIAC(3,I))
1546 11    CONTINUE
1547 12    CONTINUE
1548 C-----
1549 C-----CONTROL OF X*MAX
1550 C-----
1551      XL(IX)=AMIN1(XL(IX),XMAX)
1552      IF(ABS(XL(IX)-XMAX).LT.EPS) GOTO 16
1553 15    CONTINUE
1554 16    CONTINUE
1555 C-----
1556 C-----GENERATE Y-LINES
1557 C-----
1558      DO 20 IY=2,170
1559      YL(IY)=YL(IY-1)+YBOX
1560 C-----
1561 C-----CONTROL OF SPECIFIED Y-LINES
1562 C-----
1563      DO 17 I=1,NY
1564      IF(YL(IY-1).LT.(YA(I)-EPS)) YL(IY)=AMIN1(YL(IY),YA(I))
1565 17    CONTINUE
1566 C-----
1567 C-----CONTROL OF SUBREGION LIMITS
1568 C-----
1569      IF(NR.EQ.1) GOTO 19
1570      DO 18 I=2,NR
1571      IF(YL(IY-1).LT.(SRDIAC(2,I)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(2,I))
1572      IF(YL(IY-1).LT.(SRDIAC(4,I)-EPS)) YL(IY)=AMIN1(YL(IY),SRDIAC(4,I))
1573 18    CONTINUE
1574 19    CONTINUE
1575 C-----
1576 C-----CONTROL OF YMAX
1577 C-----
1578      YL(IY)=AMIN1(YL(IY),YMAX)
1579      IF(ABS(YL(IY)-YMAX).LT.EPS) GOTO 21
1580 20    CONTINUE
1581 21    CONTINUE
1582 C-----PRINT COORDINATES OF X - AND Y - LINES
1583 PRINT 310,IX,(XL(I),I=1,IX)
1584 PRINT 320,IY,(YL(I),I=1,IY)
1585 C-----
1586 NE=(IX-1)*(IY-1)
1587 NN=IX*IY
1588 PRINT 330,NN,NE
1589 MAX=IY+2
1590 100  FORMAT()
1591 110  FORMAT(20A4)
1592 199  FORMAT(1H1)
1593 200  FORMAT(// ' TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE' /
1594      1 ' SOLVES NON LINEAR TRANSIENT FIELD PROBLEMS'
1595      2 / ' *** TWO DIMENSIONAL VERSION ***'
1596      3 / ' PROGRAMMED BY ULF WICKSTROM' / ' LUND FEB 1979' // 1X,80(1H*) ///)
1597 205  FORMAT(/// ' TITLE OF RUN : ',20A4)
1598 210  FORMAT(/// ' GEOMETRY' / 1X,8(1H*) //
1599      1 ' MAXIMUM COORDINATES',13X,'XMAX=',G10.3,5X,'YMAX=',G10.3)

```



```

1600 220 FORMAT(' MAXIMUM ELEMENT LENGTH',10X,'XBOX=',G10.3,5X,'YBOX='
1601 1 ,G10.3)
1602 230 FORMAT(' SUBREGIONS')
1603 240 FORMAT(' NUMBER OF SUBREGIONS',I4// ' SUBREGION DIAGONAL LIMITS'//
1604 1 /4X,'XMIN',6X,'YMIN',6X,'XMAX',6X,'YMAX',6X,'FICTITIOUS AREA'//
1605 2 (1X,4G10.3,10X,L1))
1606 250 FORMAT(' COORDINATES OF SPECIFIED X - LINES')
1607 260 FORMAT(/6G10.3)
1608 270 FORMAT(' COORDINATES OF SPECIFIED Y - LINES')
1609 280 FORMAT(/6G10.3)
1610 300 FORMAT(' THIS IS AN AXISYMMETRIC PROBLEM'//6X,'INNER RADIUS ',
1611 1 'XMIN = ',G10.3)
1612 310 FORMAT(' NUMBER AND COORDINATES OF X - LINES'//I3,' - ',7G10.3/
1613 1 (6X,7G10.3))
1614 320 FORMAT(' NUMBER AND COORDINATES OF Y-LINES'//I3,' - ',7G10.3/
1615 1 (6X,7G10.3))
1616 330 FORMAT(' NUMBER OF NODES=',I4,10X,'NUMBER OF ELEMENTS=',I4)
1617 RETURN
1618 END
1619
1620 SUBROUTINE NGROUP(X,Y)
1621 C-----THIS ROUTINE READS AND FORMS NODE GROUP DATA
1622 DIMENSION X(1),Y(1)
1623 PARAMETER NB=10,NNB=20,NNB2=2*NNB
1624 COMMON/NBOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),
1625 1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
1626 COMMON/ENRAD/E(1000)
1627 COMMON/ENCON/H(50),TAIR(2)
1628 LOGICAL FA
1629
1630 C-----
1631 PRINT 200
1632 READ 100,NGROUP
1633
1634 C-----
1635 DO 10 I=1,NGROUP
1636 C-----
1637 READ 100,NCHECK,NUMB(I),EPSG(I),BETA(I),CPG(I)
1638
1639 C-----
1640 IF(I.NE.NCHECK) GO TO 1000
1641 NUMI=NUMB(I)
1642
1643 C-----
1644 READ 100,(NBOUND(I,J),J=1,NUMI)
1645
1646 C-----
1647 NOD1=NBOUND(I,1)
1648 DO 10 J=2,NUMI
1649 NOD2=NBOUND(I,J)
1650 BAREA(I,J)=SQRT((X(NOD1)-X(NOD2))**2+(Y(NOD1)-Y(NOD2))**2)
1651 NOD1=NOD2
1652 10 CONTINUE
1653
1654 C
1655 C-----PRINT INPUT DATA
1656 C
1657 DO 15 I=1,NGROUP
1658 NUMI=NUMB(I)
1659 PRINT 210,I
1660 IF (EPSG(I).EQ.C..AND.BETA(I).EQ.0.) GO TO 20
1661 PRINT 220,EPSG(I),BETA(I),CPG(I)
1662 PRINT 230,(NBOUND(I,J),J=1,NUMI)
1663 200 FORMAT(' NODE GROUPS'/1X,11(1H*))
1664 210 FORMAT(' NODE GROUP',I3)
1665 220 FORMAT(' EMISSIVITY=',G9.3/' CONVECTION FACTOR=',G9.3/
1666 1 ' CONVECTION POWER=',G9.3)
1667 230 FORMAT(' NODES',10I5/6X,10I5)
1668 15 CONTINUE
1669 GO TO 1001

```

```

1664      1000 PRINT 240
1665      STOP
1666      1001 RETURN
1667      100 FORMAT( )
1668      240 FORMAT(// ' WRONG INPUT OF NODE GROUPS ' )
1669      END
1670
1671      SUBROUTINE OUTMA2(IX,IY,NN,NE,X,Y,TIME,KTIME,T,TT,TMAX,FLOW,AXIAL)
1672      C-----THIS ROUTINE PRINTS MAXIMUM CALCULATED NODAL TEMPERATURES
1673      COMMON/FIRE/TIM(50),TB(50),TITFIR
1674      INTEGER TITFIR(18)
1675      LOGICAL TMAX,AXIAL
1676      DIMENSION X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),FLOW(NN)
1677      PRINT 200,TITFIR,X(NN),Y(NN)
1678      IDUM1=1-IY
1679      DO 10 I=1,IX
1680      IDUM1=IDUM1+IY
1681      IDUM2=IDUM1+IY-1
1682      IF(IY.LE.7) PRINT 210,(J,TT(J),J=IDUM1,IDUM2)
1683      IF(IY.GT.7) PRINT 230,(TT(J),J=IDUM1,IDUM2)
1684      10 CONTINUE
1685      PRINT 220,TIME,KTIME
1686      200 FORMAT(//1X,75(1HF)/2H,F/' F MAXIMAL TEMPERATURES/' F ' ,
1687      1 18A4/' F XMAX=' ,F8.3,1CX,'YMAX=' ,F8.3/' F ' )
1688      210 FORMAT(' F',13(1F5.0))
1689      220 FORMAT(2H F/2H F/' F MAX-TIME',F7.2,10X,'NUMBER OF '
1690      1 'TIME INCREMENTS',15/2H F/2H F/2H F,75(1HF))
1691      230 FORMAT(' F',18F7.0)
1692      RETURN
1693      END
1694
1695      SUBROUTINE OUT2(IX,IY,NN,NE,X,Y,TIME,KTIME,DELTI,T,TT,TMAX,FLOW,
1696      1 TFIRE,NODT,AXIAL)
1697      C-----THIS ROUTINE PRINTS NODAL TEMPERATURES AND VOID AIR TEMPERATURES
1698      DIMENSION X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),FLOW(NN)
1699      LOGICAL TMAX,AXIAL,LDUM,LEN
1700      COMMON/ENCON/H(50),TAIR(2)
1701      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODEN(2),INODEN(100),
1702      1 XSYM(2),YSYM(2)
1703      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
1704      TIME1=TIME-DELTI
1705      DO 5 IJ=1,NN
1706      IF(TMAX(IJ)) GOTO 5
1707      C-----IF THE NODAL TEMPERATURE DECREASES SET TMAX=.TRUE. AND PRINT
1708      C-----MAX TEMPERATURE TT
1709      IF(TT(IJ).GT.1.001*T(IJ))
1710      1 PRINT 200,IJ,TT(IJ),TIME1,DELTI
1711      IF(TT(IJ).GT.1.001*T(IJ)) TMAX(IJ)=.TRUE.
1712      TT(IJ)=AMAX1(TT(IJ),T(IJ))
1713      5 CONTINUE
1714      C-----IF TIME=TOUT PRINT ALL TEMPERATURES
1715      IF((TIME-TOUT(II)).LT.-1.E-4) GOTO 70
1716      PRINT 100,TIME,KTIME,TFIRE,NODT
1717      IF(.NOT.LEN) GOTO 30
1718      PRINT 300
1719      DO 20 I=1,NENC
1720      PRINT 310,I,TAIR(I)
1721      30 CONTINUE
1722      II=II+1
1723      IDUM1=1-IY
1724      LDUM=IY.LT.7
1725      DO 10 I=1,IX
1726      IDUM1=IDUM1+IY
1727      IDUM2=IDUM1+IY-1

```

```

1728      IF(LDUM) PRINT 210,(J,T(J),J=IDUM1,IDUM2)
1729      IF(.NOT.LDUM) PRINT 220,(T(J),J=IDUM1,IDUM2)
1730      10 CONTINUE
1731      70 CONTINUE
1732      100 FORMAT(///1X,6(1H*),6H TIME,F8.3,2X,6(1H*),' INCREMENT',
1733      1 ' NUMBER',I6,
1734      22X,17(1H*)// ' FIRE TEMPERATURE',F7.0,2X,6(1H*)
1735      3,2X,' TIME INCREMENT LIMITING NODE',I5/)
1736      200 FORMAT(///4H NOD,I4,5X,8HMAX TEMP,F6.0,5X,'TIME',G10.4,5X,
1737      1 SHDELT,I,G10.4/1X,70(1H*))
1738      210 FORMAT(13(I5,F5.0))
1739      220 FORMAT(18F7.0)
1740      300 FORMAT(' ENCLOSURE AIR TEMPERATURE')
1741      310 FORMAT(' VOID NUMBER',I2,' TAIR=',F5.0/)
1742      RETURN
1743      END
1744
1745      SUBROUTINE PPOG2(IX,IY,NN,NE,NR,N,KTOP,NODEL,MNODEL,X,Y,T,TT,TMAX,
1746      1 ELA,EV4,A,MAX,P,W,EN,F,FLOW,AXIAL,NODCPL,NODINT,DTA)
1747      C-----
1748      C-----
1749      C----- *** T A S E F ***
1750      C-----
1751      C-----TEMPERATURE ANALYSIS OF STRUCTURES EXPOSED TO FIRE
1752      C-----
1753      C-----FINITE ELEMENT PROGRAM FOR ANALYSIS OF TRANSIENT NONLINEAR
1754      C-----HEAT TRANSFER PROBLEMS
1755      C-----
1756      C-----PROGRAMMED BY
1757      C-----ULF WICKSTROM
1758      C-----LUND INSTITUTE OF TECHNOLOGY
1759      C-----MARCH 1979
1760      C-----
1761      C-----
1762      C-----THIS IS THE MAIN CONTROL ROUTINE
1763      C-----
1764      C-----DEFINITIONS OF VARIABLES
1765      C-----IX,IY      NUMBER OF X- AND Y-LINES
1766      C-----NN      NUMBER OF NODES IN BASE STRUCTURE
1767      C-----NE      NUMBER OF ELEMENTS IN BASE STRUCTURE
1768      C-----NR      NUMBER OF REGIONS
1769      C-----N      VECTOR OF REGION NUMBERS
1770      C-----KTOP     NODES ADJACENT TO EACH ELEMENT
1771      C-----NODEL     ELEMENTS ADJACENT TO EACH NODE
1772      C-----MNODEL    NUMBER ELEMENTS ADJACENT TO EACH NODE
1773      C-----X,Y      NODE COORDINATES
1774      C-----T      CURRENT NODAL TEMPERATURES
1775      C-----TT      MAXIMUM NODAL TEMPERATURES
1776      C-----TMAX     TRUE IF MAXIMUM NODAL TEMPERATURE OBTAINED
1777      C-----ELA,EV4   DUMMY GEOMETRICAL CONSTANTS
1778      C-----A      HEAT CONDUCTION MATRIX
1779      C-----P      HEAT CAPACITY VECTOR
1780      C-----W      NODAL VOLUME VECTOR
1781      C-----EN      NODAL ENTHALPY VECTOR
1782      C-----F      INTERNAL NODAL HEAT FLOW VECTOR
1783      C-----FLOW     EXTERNAL NODAL HEAT FLOW VECTOR
1784      C-----AXIAL    TRUE IF AXISYMMETRIC PROBLEM
1785      C-----NODCPL   INDICATES COUPLED NODES
1786      C-----NODINT   INDICATES INTERFACE NODES
1787      C-----DTA      DUMMY VECTOR FOR CRITICAL TIME INCREMENT CALCULATION
1788      C-----
1789      C-----PARAMETER CONSTANTS
1790      C-----
1791      C-----NB      MAXIMUM NUMBER OF NODE GROUPS

```

```

1792 C-----NNR          MAXIMUM NUMBER OF NODES IN ONE NODE GROUP
1793 C-----NCP          MAXIMUM NUMBER OF COUPLED GROUPS OF NODES
1794 C-----MNR          MAXIMUM NUMBER OF REGIONS
1795 C-----MNV          MAXIMUM NUMBER OF VALUE PAIRS
1796 C-----
1797 C-----COMMON FIELDS
1798 C-----
1799 C-----COUPLE        DATA ON COUPLED NODES
1800 C-----      NCPLG      NUMBER OF COUPLED GROUPS OF NODES
1801 C-----      NCOUPL     MATRIX OF COUPLED NODES
1802 C-----DIM          DIMENSIONS OF CERTAIN ARRAYS
1803 C-----      MAXNG      MAXIMUM NUMBER OF NODE GROUPS DEFINING ONE ENCLOSURE
1804 C-----      MAXNOD     MAXIMUM NUMBER OF NODES AROUND ONE ENCLOSURE
1805 C-----DUMMY        DUMMY MATRICES
1806 C-----ENCLOS      ENCLOSURE DATA
1807 C-----      LEN        TRUE IF STRUCTURE CONTAINS VOID OR ENCLOSURE
1808 C-----      NENC       NUMBER OF ENCLOSURES
1809 C-----      NENCNG     VECTOR OF NUMBER OF NODE GROUPS
1810 C-----      IGREN      MATRIX OF NODES
1811 C-----      NNODFN     NUMBER OF NODES SURROUNDING AVOID
1812 C-----      XSYM       TRUE IF VOID SYMMETRICAL AROUND X-AXIS
1813 C-----      YSYM       TRUE IF VOID SYMMETRICAL AROUND Y-AXIS
1814 C-----ENCON        ENCLOSURE CONVECTION DATA
1815 C-----      H          ARRAY OF ENCLOSURE CONVECTION VECTORS
1816 C-----      TAIR       ENCLOSURE AIR TEMPERATURE
1817 C-----ENRAD        ENCLOSURE RADIATION DATA
1818 C-----      E          ARRAY OF ENCLOSURE RADIATION MATRICES
1819 C-----FIRE          FIRE TEMPERATURE DATA
1820 C-----      TIM,TR      TIME - FIRE TEMPERATURE PAIRS
1821 C-----      TITFIR     FIRE IDENTIFIER
1822 C-----FOB          PRESCRIBED HEAT FLOW DATA
1823 C-----      NFQNG      NUMBER OF NODE GROUPS DEFINING PRESCRIBED FLOW BOUNDARIES
1824 C-----      NFQG       VECTOR OF NODE GROUPS DEFINING PRESCRIBED FLOW
1825 C-----      TR,TC      VECTORS OF MODIFIED TEMPERATURE
1826 C-----      BR,BC      RADIATION AND CONVECTION BOUNDARY MATRICES
1827 C-----BNOD        DATA ON NODE GROUPS
1828 C-----      NUMR       VECTOR OF NUMBER OF NODES IN THE NODE GROUPS
1829 C-----      NBOUND     MATRIX OF NODE NUMBERS IN THE NODE GROUPS
1830 C-----      BAREA      MATRIX OF DISTANCES BETWEEN NODES
1831 C-----      EPSG       VECTOR OF EMISSIVITY OF NODE GROUPS
1832 C-----      BFTA       VECTOR OF CONVECTION FACTORS OF NODE GROUPS
1833 C-----      CPG        VECTOR OF CONVECTION POWERS OF NODE GROUPS
1834 C-----      FA         TRUE FOR FIRE BOUNDARY NODE GROUPS
1835 C-----PTB          PRESCRIBED TEMPERATURE
1836 C-----      NPTNG      NUMBER OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURE
1837 C-----      NPTG       VECTOR OF NODE GROUPS DEFINING PRESCRIBED TEMPERATURES
1838 C-----RGE0        GEOMETRIC DATA
1839 C-----      ELFICT     TRUE FOR FICTITIOUS ELEMENTS
1840 C-----      ET         ELEMENT THICKNESS
1841 C-----      SRDIAC     SUBREGION DIAGONAL DATA
1842 C-----RMAT        MATERIAL DATA
1843 C-----      CCC        TRUE IF CONDUCTIVITY IS FUNCTION MAXIMUM TEMPERATURE
1844 C-----      TC,C       TEMPERATURE - CONDUCTIVITY PAIRS
1845 C-----      TF,ENT     TEMPERATURE - SPECIFIC VOLUMETRIC ENTHALPY PAIRS
1846 C-----      CR         NOMINAL SPECIFIC VOLUMETRIC HEAT
1847 C-----      TQ,QE      TEMPERATURE - INTERNALLY GENERATED HEAT PAIRS
1848 C-----      LQ         TRUE IF INTERNAL HEAT IS GENERATED
1849 C-----TOUT        TIME DATA
1850 C-----      II         COUNTER
1851 C-----      TOUT       VECTOR OF PRINT OUT TIMES
1852 C-----      TIMMAX     MAXIMUM TIME
1853 C-----      DTMAX      MAXIMUM TIME INCREMENT
1854 C-----      TIMFAC     TIME INCREMENT FACTOR
1855 C-----      KTMAX      MAXIMUM NUMBER TIME INCREMENTS

```

```

1856 C----- KUPDA      NUMBER OF TIME STEPS BETWEEN UPDATING CONDUCTION MATRIX
1857 C-----UNIT      UNIT DEPENDENT CONSTANTS
1858 C----- SIGMA      STEFAN-BOLTZMANN CONSTANT
1859 C----- TABS       ABSOLUTE TEMPERATURE SHIFT
1860 C----- TINIT      INITIAL TEMPERATURE
1861 C-----
1862      DIMENSION N(NE),KTOP(4,NE),X(NN),Y(NN),T(NN),TT(NN),TMAX(NN),
1863      1 ELA(4,NE),EV4(NF),A(NN,MAX),P(NN),FN(NN),F(NN),FLOW(NN),
1864      2 NODCPL(NN),NODINT(NN),W(NN),NODEL(4,NN),MNODEL(NN),DTA(NN)
1865      LOGICAL TMAX,AXIAL
1866 C-----FORM THE VECTOR N
1867      CALL REG2(NN,NE,NR,N,KTOP,X,Y,NODEL,MNODEL)
1868 C-----DEFINE COUPLED NODES
1869      CALL COUPLA(NODCPL,NN,NODINT)
1870 C-----DEFINE INTERFACE NODES
1871      CALL INTERF(NN,NF,NR,IX,IY,KTOP,N,NODINT,NODCPL)
1872 C-----INPUT MATERIAL DATA
1873      CALL MAT(NR)
1874 C-----FORM GEOMETRICAL DUMMY CONSTANTS
1875      CALL GEOCO2(NN,NE,N,KTOP,X,Y,AXIAL,ELA,EV4)
1876 C-----INPUT INITIAL DATA
1877      CALL AMB
1878 C-----FORM NODE GROUPS
1879      CALL NGROUP(X,Y)
1880 C-----DEFINE FIRE PRESCRIBED HEAT FLOW BOUNDARIES
1881      CALL FGBNDA
1882 C-----DEFINE FIRE PRESCRIBED TEMPERATURE BOUNDARIES
1883      CALL PTBND4(NODCPL)
1884 C-----DEFINE ENCLOSURE BOUNDARIES
1885      CALL ENCL01(X,Y)
1886 C-----INPUT TIME DATA
1887      CALL TIME
1888 C-----FORM NODE VOLUME VECTOR
1889      CALL ASSW2(NN,NF,N,KTOP,X,Y,EV4,AXIAL,W)
1890 C-----SUMMERIZE APPROPRIATE NODE VOLUMES
1891      CALL COUPLB(W)
1892 C-----CALL TIME INTEGRATION CONTROL ROUTINE
1893      CALL FEM2(IX,IY,NN,NE,NP,N,KTOP,X,Y,T,TT,TMAX,ELA,EV4,A,MAX,P,W,
1894      1 EN,F,FLOW,AXIAL,NODCPL,NODINT,NODEL,MNODEL,DTA)
1895      RETURN
1896      END
1897
1898      SUBROUTINE PTBND4(NODCPL)
1899 C-----INPUT NODE GROUPS OF PRESCRIBED TEMPERATURE
1900      DIMENSION NODCPL(1)
1901      PARAMETER NR=10,NNR=30,NNR2=2*NNR
1902      COMMON/PTR/NPTNG,NPTG(NB)
1903      COMMON/BNOD/NUMB(NR),NBOUND(NB,NNR),BARFA(NB,NNR),TH(NR),
1904      1 EPSG(NB),BETA(NB),CPG(NB),FA(NB)
1905      LOGICAL FA,FA1
1906 C-----READ NUMBER OF BOUNDARY NODES GROUPS
1907      READ 100,NPTNG
1908      IF(NPTNG.EQ.0) RETURN
1909      PRINT 200
1910 C-----EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
1911      DO 20 IB=1,NPTNG
1912      C-----
1913      READ 100,FA1,ING1
1914 C-----FA1 = TRUE FIRE BOUNDARY ELSE AMBIENT TEMPERATURE
1915 C-----ING1 = NODE GROUP NUMBER
1916      FA(ING1)=FA1
1917      NUMI=NUMB(ING1)
1918      NPTG(IB)=ING1
1919      DO 10 J=1,NUMI

```

```

1920      NOD=NBOUND(ING1,J)
1921      IF (NODCPL(NOD).EQ.0) PRINT 300,NOD
1922      IF (NODCPL(NOD).EQ.0) STOP
1923      10  NODCPL(NOD)=0
1924          IF(FA1) PRINT 210,ING1
1925          IF(.NOT.FA1) PRINT 220,ING1
1926      20  CONTINUE
1927      100  FORMAT()
1928      210  FORMAT(' NODE GROUP',I3,' FIRE BOUNDARY')
1929      220  FORMAT(' NODE GROUP',I3,' AMBIENT BOUNDARY')
1930      200  FORMAT(/' PRESCRIBED TEMPERATURE BOUNDARY'/1X,31(1H*))//
1931      1  ' NODE GROUPS AND TYPES OF BOUNDARIES')
1932      300  FORMAT(/' NODE',I4,' IS A SLAVE NODE')
1933      1  ' SLAVE NODES CANNOT HAVE PRESCRIBED TEMPERATURE')
1934      RETURN
1935      END
1936
1937      SUBROUTINE PTBND(B,T,TFIRE)
1938      C-----SET PRESCRIBED NODAL BOUNDARY TEMPERATURE
1939      DIMENSION T(1)
1940      PARAMETER NB=10,NNR=30,NNB2=2*NNB
1941      COMMON/PTB/NPTNG,NPTG(NB)
1942      COMMON/BNOD/NUMB(NB),NBOUND(NB,NNB),BAREA(NB,NNB),TH(NB),
1943      1  EPSG(NB),BETA(NB),CPG(NB),FA(NB)
1944      COMMON/UNIT/SIGMA,TABS,TINIT,TAMB,TAMB4
1945      LOGICAL FA
1946      C-----
1947      IF(NPTNG.EQ.0) RETURN
1948      C-----EACH PRESCRIBED TEMPERATURE BOUNDARY NODE GROUP
1949      DO 10 IB=1,NPTNG
1950          TG=TAMB
1951          ING1=NPTG(IB)
1952          IF(FA(ING1)) TG=TFIRE
1953          NUMI=NUMB(ING1)
1954          DO 10 I=1,NUMI
1955              NODE=NBOUND(ING1,I)
1956              T(NODE)=TG
1957      10  CONTINUE
1958      RETURN
1959      END
1960
1961      SUBROUTINE RADVEC(F,ETA,N,Q)
1962      C-----THIS ROUTINE FORMS THE LOCAL ENCLOSURE SURFACE RADIATION HEAT
1963      C-----EXCHANGE VECTOR Q=F*ETA
1964      DIMENSION Q(1),ETA(1),E(N,N)
1965      QTOT=0.
1966      DO 20 I=1,N
1967          QT=0.
1968          DO 10 J=1,N
1969              QT=QT+E(I,J)*ETA(J)
1970          QTOT=QTOT+QT
1971      20  Q(I)=QT
1972      RETURN
1973      220  FORMAT(/' TOTAL RADIATION HEAT EXCHANGE',E11.3)
1974      END
1975
1976      SUBROUTINE REG2(NN,NE,NR,N,KTOP,X,Y,NODEL,MNODEL)
1977      C-----THIS SUBROUTINE FORMS VECTOR OF REGION NUMBERS N OF EACH ELEMENT
1978      DIMENSION X(NN),Y(NN),N(NE),KTOP(4,NE),NODEL(4,NN),MNODEL(NN)
1979      PARAMETER MNR=10
1980      COMMON/RGEO/ELFICT(MNR),ET(MNR),SRDIAC(4,MNR)
1981      LOGICAL ELFICT
1982      EPS=1.E-7
1983      DO 10 I=1,NE

```



```

1984      N(I)=1
1985      IF(NR.EQ.1) GOTO 10
1986      ND1=KTOP(1,I)
1987      ND2=KTOP(4,I)
1988      DO 5 J=2,NR
1989      IF((X(ND1)-SRDIAC(3,J)).GT.-EPS) GOTO 5
1990      IF((Y(ND1)-SRDIAC(4,J)).GT.-EPS) GOTO 5
1991      IF((X(ND2)-SRDIAC(1,J)).LT.EPS) GOTO 5
1992      IF((Y(ND2)-SRDIAC(2,J)).LT.EPS) GOTO 5
1993      N(I)=J
1994      5 CONTINUE
1995      10 CONTINUE
1996      DO 40 I=1,NN
1997      II=0
1998      DO 30 IE=1,NE
1999      N1=N(IE)
2000      IF(ELFICT(N1)) GOTO 30
2001      DO 20 J=1,4
2002      IF(KTOP(J,IE).NE.I) GOTO 20
2003      II=II+1
2004      NODEL(II,I)=IE
2005      IF(II.EQ.4) GOTO 30
2006      20 CONTINUE
2007      30 CONTINUE
2008      MNODEL(I)=II
2009      40 CONTINUE
2010      RETURN
2011      END
2012
2013      SUBROUTINE TIME
2014      C-----READ TIME INTEGRATION CONTROL DATA
2015      COMMON/TOUT/II,TOUT(100),TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
2016      PRINT 200
2017      C-----
2018      READ 100,NT,TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
2019      C-----
2020      IF(DTMAX.EQ.0) DTMAX=TIMMAX
2021      IF(TIMFAC.EQ.0) TIMFAC=.8
2022      IF(KTMAX.EQ.0) KTMAX=1000
2023      IF(KUPDA.EQ.0) KUPDA=1
2024      C-----
2025      READ 100,(TOUT(I),I=1,NT)
2026      C-----
2027      PRINT 210,TIMMAX,DTMAX,TIMFAC,KTMAX,KUPDA
2028      PRINT 220,(TOUT(I),I=1,NT)
2029      100 FORMAT()
2030      220 FORMAT(' PRINT OUT TIMES',3X,867.2/(19X,867.2))
2031      200 FORMAT(//' TIME'/' ****'/)
2032      210 FORMAT(' MAXIMUM TIME=',G8.3/' MAXIMUM TIME INCRFMNT=',G8.3/
2033      1 ' CRITICAL TIME INCREMENT FACTOR=',G8.3/
2034      2 ' MAXIMUM NUMBER OF TIME INCREMENTS=',I5/
2035      3 ' NUMBER OF STEPS BETWEEN UPDATING OF CONDUCTION MATRIX=',I5)
2036      RETURN
2037      END
2038
2039      SUBROUTINE VIEWFC(X,Y,D,EN,VIEW,MAXNOD)
2040      C-----THIS ROUTINE CALCULATES VIEW-FACTORS AND ENCLOSURE ZONE AREAS
2041      C-----SYMMETRY AROUND ANY OR BOTH AXIS ARE TAKEN INTO ACCOUNT
2042      DIMENSION X(1),Y(1),D(1),VIEW(MAXNOD,MAXNOD)
2043      PARAMETER NB=10,NNR=30,NNR2=2*NNB
2044      COMMON/BNOD/NUMB(NR),NBOUND(NR,NNR),BAREA(NB,NNR),TH(NR),
2045      1 EPSG(NB),RETA(NB),CPG(NB),FA(NB)
2046      COMMON/ENCLOS/LEN,NENC,NENCNG(2),IGREN(2,4),NNODFN(2),TMODFN(100),
2047      1 XSYM(2),YSYM(2)

```



```

2048          LOGICAL LFN
2049          LOGICAL LDUM,XSYM,YSYM,DSYM
2050          INTEGER EN
2051          DO 15 I=1,MAXNOD
2052          DO 15 J=1,MAXNOD
2053          15 VIEW(I,J)=0.
2054          C-----COMPUTE VIEW-FACTORS USING HOTTEL'S CROSSED-STRING METHOD
2055          NENG=NENCMG(FN)
2056          SIGNX2=1.
2057          SIGNY2=1.
2058          IN=0
2059          C-----EACH NODE GROUP
2060          DO 100 IG=1,NENG
2061          DSYM=.FALSE.
2062          IF(XSYM(EN).AND.YSYM(EN)) DSYM=.TRUE.
2063          I1=IGREN(EN,IG)
2064          NUMI=NUMB(I1)
2065          C-----EACH ZONE
2066          DO 100 I=2,NUMI
2067          IN=IN+1
2068          NOD1=NBOUND(I1,I-1)
2069          NOD2=NBOUND(I1,I)
2070          X1=X(NOD1)
2071          X2=X(NOD2)
2072          Y1=Y(NOD1)
2073          Y2=Y(NOD2)
2074          D1=BAREA(I1,I)
2075          C-----FORM THE ZONE AREA VECTOR D
2076          D(IN)=D1
2077          JN=0
2078          C-----EACH NODE GROUP
2079          DO 100 JG=1,NENG
2080          J1=IGREN(EN,JG)
2081          NUMJ=NUMB(J1)
2082          C-----EACH ZONE
2083          DO 100 J=2,NUMJ
2084          JN=JN+1
2085          D2=BAREA(J1,J)
2086          NOD3=NBOUND(J1,J-1)
2087          NOD4=NBOUND(J1,J)
2088          IF(XSYM(EN)) SIGNY2=-1.
2089          IF(YSYM(EN)) SIGNX2=-1.
2090          50 CONTINUE
2091          LDUM=SIGNX2.EQ.1..AND.SIGNY2.EQ.1.
2092          IF(.NOT.LDUM) GOTO 80
2093          IF(IN.GE.JN) GOTO 100
2094          80 CONTINUE
2095          X3=SIGNX2*X(NOD3)
2096          X4=SIGNX2*X(NOD4)
2097          Y3=SIGNY2*Y(NOD3)
2098          Y4=SIGNY2*Y(NOD4)
2099          D3=SQRT((X1-X3)**2+(Y1-Y3)**2)
2100          D4=SQRT((X2-X4)**2+(Y2-Y4)**2)
2101          D5=SQRT((X1-X4)**2+(Y1-Y4)**2)
2102          D6=SQRT((X2-X3)**2+(Y2-Y3)**2)
2103          DUM=ABS(D5+D6-D4-D3)/2.
2104          C-----HOTTEL'S CROSSED-STRING METHOD
2105          VIEW(IN,JN)=DUM/D1+VIEW(IN,JN)
2106          C-----TAKE ADVANTAGE OF RECIPROCITY
2107          IF(LDUM) VIEW(JN,IN)=DUM/D2
2108          C-----IF SYMMETRY AROUND AXIS GO BACK WITH CHANGED SIGNS OF COORDINATES
2109          LDUM=(SIGNX2+SIGNY2).LT.-1.99
2110          IF(LDUM) SIGNY2=1.
2111          IF(LDUM) GOTO 50

```

```

2112 C-----
2113 LDUM=DSYM.AND.SIGNX2.LT.0.
2114 IF(LDUM) SIGNX2=1.
2115 IF(LDUM) SIGNY2=-1.
2116 IF(LDUM) GOTO 50
2117 C-----
2118 LDUM=(SIGNX2*SIGNY2).LT.0.
2119 IF(LDUM) SIGNX2=1.
2120 IF(LDUM) SIGNY2=1.
2121 IF(LDUM) GOTO 50
2122 100 CONTINUE
2123 RETURN
2124 END
2125
2126 SUBROUTINE XVERPSY(X,Y,N,M,XS,YS)
2127 C-----FIND YS AS FUNCTION OF XS BY LINEAR INTERPOLATION
2128 C-----IN TABLE OF X- AND Y-VALUES
2129 DIMENSION Y(N,1),X(N,1)
2130 DO 10 I=2,N
2131 IF(XS.GE.X(I,M))GOTO 10
2132 YS=Y(I-1,M)+(XS-X(I-1,M))*(Y(I,M)-Y(I-1,M))/(X(I,M)-X(I-1,M))
2133 GOTO 11
2134 10 CONTINUE
2135 PRINT 1
2136 PRINT 2,XS,(X(I,M),Y(I,M),I=1,5)
2137 PRINT 3,M
2138 STOP
2139 1 FORMAT(///1X,'INPUT VALUE TO XVERSY OUT OF RANGE!')
2140 2 FORMAT(1X,'INPUT X',E10.4/1X'X-Y VALUE PAIRS'/1X,5(2G9.3,5X))
2141 3 FORMAT(/1X,'CURVE NUMBER',I4)
2142 11 RETURN
2143 END

```

APPENDIX C - Example input

Input cards used in examples I-III in Section 5.

Example I

```
1    SQUARE PLATE
2    F,1,1,.125,.125,,,
3    0
4    UNIT MATERIAL DATA
5    F,2,2,,,
6    ,1,10000,1,
7    ,10000,10000,
8    .00001,1000,,,
9    1
10   1,17,,1,1,
11   9 18 27 36 45 54 63 72 81 80 79 78 77 76 75 74 73
12   1
13   F,1
14   0,
15   NOVOID
16   20,1,1,1,,,
17   .05 .10 .15 .20 .25 .30 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .85 .90 .95 1.
18   DUMMY TEMPERATURE
19   2
20   ,,1000,,
21
```

Example II

```

1      I BEAM EMBEDDED IN CONCRETE
2      F,.14,.1,.05,.03,6,4,4
3      F,,,05,.01
4      F,,,01,.003,.09
5      F,,,09,.05,.1
6      T,.05,,,14,.01
7      T,.05,.09,.14,.1
8      .015,.03,.07,.1,
9      .02,.035,.05,.08
10     12
11     33 34,,,,,,
12     25 26,,,,,,
13     17 18,,,,,,
14     1 2 9 10,,,,,,
15     3 11,,,,,,
16     4 12,,,,,,
17     5 13,,,,,,
18     6 14,,,,,,
19     7 15 8 16,,,,,,
20     23 24,,,,,,
21     31 32,,,,,,
22     39 40,,,,,,
23     BETONG
24     T,7,7,,
25     24.5,1.78,115,1.28,243,1.17,401,1.17,643,.92,895,.85,1500,.85,
26     ,100,55600,115,91000,200,129400,600,397200,1000,696700,1500,1000000.,
27     STAL
28     F,3,7,,
29     ,60,800,27,2000,27
30     ,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
31     STAL
32     F,3,7,,
33     ,60,800,27,2000,27
34     ,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
35     STAL
36     F,3,7,,
37     ,60,800,27,2000,27
38     ,200.,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
39     25.0001,25.,,,,
40     4
41     1,6,.6,.99,1.33
42     1,9,17,25,33,34,
43     2,4,.8,.99,1.33
44     34 42 50 58
45     3 6 .6 2.2,1.25
46     8 16 24 32 40 39
47     4 4 .8 2.2,1.25
48     39 47 55 63
49     4
50     T,1
51     T,2
52     F,3
53     F,4
54     0
55     NOVOID
56     15,1.5,,,,,,
57     .1,.2,.3,.4,.5,.6,.7,.8,.9,1.0,1.1,1.2,1.3,1.4,1.5,
58     HE100B#FIRE
59     9
60     ,25,.05,525,.1,620,.3,725,.6,940,1,980.,1.025,600,1.2,475,1.5001,360,,,,
61

```

Example III

```

1 BOX GIRDER EMBEDDED IN CONCRETE
2 F,.14,.15,1.,1.,5,4,6
3 F,,,0875,.004
4 F,.0345,.004,.0375,.112
5 F,.112,.0375,.12
6 T,.004,.0345,.112
7 .02,.05,.07,.11
8 .015,.029,.044,.056,.085,.13
9 14
10 1 2,,,,,,
11 12 13,,,,,,
12 23 24 34 35,,,,,,
13 45 46,,,,,,
14 56 57,,,,,,
15 68 67,,,,,,
16 25 36,,,,,,
17 26 37,,,,,,
18 27 38,,,,,,
19 28 39,,,,,,
20 29 40,,,,,,
21 30 31 41 42,,,,,,
22 19 20,,,,,,
23 8 9,,,,,,
24 BETONG
25 T,7,7,,
26 24,5,1.78,115,1.28,243,1.17,401,1.17,643,.92,895,.85,1500,.85,
27 ,100,55600,115,91000,200,129400,600,397200,1000,696700,1500,1000000,,
28 STAL
29 F,3,7,,
30 ,60,800,27,2000,27
31 ,200,,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
32 STAL
33 F,3,7,,
34 ,60,800,27,2000,27
35 ,200,,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
36 STAL
37 F,3,7,,
38 ,60,800,27,2000,27
39 ,200,,217000,400,466000,600,758000,700,927000,800,1192000,1200,1766000
40 25,25.0001,,
41 4
42 1,7,.6,.99,1.33
43 1,12,23,34,45,56,67,
44 2,3,.8,.99,1.33
45 67,78,89
46 3,9,.8,2,20,1.25
47 11,22,33,44,55,66,77,88,99,
48 4,11,.6,1.6,1.33
49 2,13,24,25,26,27,28,29,30,19,8
50 3
51 T,1
52 T,2
53 F,3
54 0, PTBND A
55 VOID
56 1
57 F,T,4,,,,
58 15,1.5,,,,,,
59 .1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5,
60 HSQ-FIRE
61 12
62 ,25,.05,525,.1,625,.2,765,.3,900,.5,925,.7,980,1,1015,1.05,675,1.1,600,1.2,500.
63 1.501,350.,

```